Brocade Communications Systems, Inc. Brocade MLXe® and NetIron® Family Devices with Multi-Service IronWare R05.8.00 Security Target

Version 0.4
March 31, 2015

Prepared for:

Brocade Communications Systems, Inc.

130 Holger Way
San Jose, CA 95134

Prepared By:

Gossamer Laboratories

LIST OF TABLES

1. Security Target Introduction

This section identifies the Security Target (ST) and Target of Evaluation (TOE) identification, ST conventions, ST conformance claims, and the ST organization. The TOE is the Brocade Communications Systems, Inc. Brocade MLXe® and NetIron® Family Devices with Multi-Service IronWare R05.8.00.

The Security Target contains the following additional sections:

- Conformance Claims (Section 2)
- Security Objectives (Section 3)
- Extended Components Definition (Section 4)
- Security Requirements (Section 5)
- TOE Summary Specification (Section 6)

Conventions

The following conventions have been applied in this document:

- Security Functional Requirements – Part 2 of the CC defines the approved set of operations that may be applied to functional requirements: iteration, assignment, selection, and refinement.
  - Iteration: allows a component to be used more than once with varying operations. In the ST, iteration is indicated by a letter placed at the end of the component. For example FDP_ACC.1a and FDP_ACC.1b indicate that the ST includes two iterations of the FDP_ACC.1 requirement, a and b.
  - Assignment: allows the specification of an identified parameter. Assignments are indicated using bold and are surrounded by brackets (e.g., [**assignment**]). Note that an assignment within a selection would be identified in italics and with embedded bold brackets (e.g., [***[selected-assignment]***]).
  - Selection: allows the specification of one or more elements from a list. Selections are indicated using bold italics and are surrounded by brackets (e.g., [***selection***]).
  - Refinement: allows the addition of details. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "… ~~some~~ **big** things …").
- The NDPP uses an additional convention – the 'case' – which defines parts of an SFR that apply only when corresponding selections are made or some other identified conditions exist. Only the applicable cases are identified in this ST and they are identified using **bold** text.
- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.

Terminology

User	Any entity (human or otherwise) outside the TOE that interacts with the TOE.
Unauthorized User	An entity that interacts (or attempts to interact) with the TOE Security Function (TSF) in an unapproved manner.
Authorized Administrator	A role with which a trusted TOE user is associated to administer both the functionality and security parameters of the TOE and its operational Environment. Such users are trusted not to compromise the security policy enforced by the TOE.

TOE User	Any person who interacts with the TOE.
External IT entity	Any IT product or system, untrusted or trusted, outside of the TOE that interacts with the TOE.
Role	A predefined set of rules establishing the allowed interactions between a user and the TOE.
Identity	A representation (e.g., a string) uniquely identifying an authorized user, which can either be the full or abbreviated name of that user or a pseudonym.
Authentication data	Information used to verify the claimed identity of a user.
Object	An entity within the TOE Security Function (TSF) Scope of Control (TSC) that contains or receives information and upon which subjects perform operations.
Subject	An entity within the TSC that causes operations to be performed.
Authorized User	A user who may, in accordance with the TOE Security Policy (TSP), perform an operation.

1.1 Security Target Reference

ST Title – Brocade Communications Systems, Inc. Brocade MLXe® and NetIron® Family Devices with Multi-Service IronWare R05.8.00 Security Target

ST Version – Version 0.4

ST Date – March 31, 2015

1.2 TOE Reference

TOE Identification – Brocade Communications Systems, Inc. Brocade MLXe® and NetIron® Family Devices with Multi-Service IronWare R05.8.00, including the following series and models

- Brocade NetIron MLXe Series Hardware Platforms (BR-MLXE-16-MR2-M-AC, BR-MLXE-16-MR2-M-DC, BR-MLXE-8-MR2-M-AC, BR-MLXE-8-MR2-M-DC, BR-MLXE-4-MR2-M-AC, and BR-MLXE-4-MR2-M-DC);
- Brocade NetIron CER 2000 Series Hardware Platforms (NI-CER-2024C-ADVPREM-AC, NI-CER-2024C-ADVPREM-DC, NI-CER-2024F-ADVPREM-AC, NI-CER-2024F-ADVPREM-DC, NI-CER-2048C-ADVPREM-AC, NI-CER-2048C-ADVPREM-DC, NI-CER-2048CX-ADVPREM-AC, NI-CER-2048CX-ADVPREM-DC, NI-CER-2048F-ADVPREM-AC, NI-CER-2048F-ADVPREM-DC, NI-CER-2048FX-ADVPREM-AC, NI-CER-2048FX-ADVPREM-DC, BR-CER-2024C-4X-RT-AC, BR-CER-2024C-4X-RT-DC, BR-CER-2024F-4X-RT-AC, and BR-CER-2024F-4X-RT-DC); and
- Brocade NetIron CES 2000 Series Hardware Platforms (BR-CES-2024C-4X-AC, BR-CES-2024C-4X-DC, BR-CES-2024F-4X-AC, and BR-CES-2024F-4X-DC).

TOE Developer – Brocade Communications Systems, Inc.

1.3 TOE Overview

The Target of Evaluation (TOE) is the Brocade MLXe® and NetIron® Family Devices with Multi-Service IronWare R05.8.00 family of products.

The TOE is composed of a hardware appliance with embedded software installed on a management processor. The embedded software is a version of Brocades' proprietary Multi-Service IronWare software. The software controls the switching and routing network frames and packets among the connections available on the hardware appliances.

All TOE appliances are configured at the factory with default parameters to allow immediate use of the system's basic features through its Command Line Interface (CLI). However, the product should be configured in accordance with the evaluated configuration (using the Multi-Service IronWare Federal Information Processing Standards and Common Criteria Guide) prior to being placed into operation. The CLI is a text based interface which is accessible from a directly connected terminal or via a remote terminal using SSH. Once configured, the MLX TOE series also offers an encrypted Web Management Interface using TLS. All of the remote management interfaces are protected using encryption as explained later in this ST.

The hardware platforms that support the TOE have a number of common hardware characteristics:

- Central processor that supports all system operations, i.e. PowerPC etc.
- Dynamic memory, used by the central processor for all system operations
- Non-volatile flash memory, used to store the operating system image, startup configuration and other relevant files.
- Multiple physical network interfaces either fixed in configuration or removable as in a chassis based product.

1.4 TOE Description

The Target of Evaluation (TOE) is Brocade Communications Systems, Inc. Brocade MLXe® and NetIron® Family Devices with Multi-Service IronWare R05.8.00, including the following series and models

- Brocade NetIron MLXe Series Hardware Platforms (BR-MLXE-16-MR2-M-AC, BR-MLXE-16-MR2-M-DC, BR-MLXE-8-MR2-M-AC, BR-MLXE-8-MR2-M-DC, BR-MLXE-4-MR2-M-AC, and BR-MLXE-4-MR2-M-DC);
- Brocade NetIron CER 2000 Series Hardware Platforms (NI-CER-2024C-ADVPREM-AC, NI-CER-2024C-ADVPREM-DC, NI-CER-2024F-ADVPREM-AC, NI-CER-2024F-ADVPREM-DC, NI-CER-2048C-ADVPREM-AC, NI-CER-2048C-ADVPREM-DC, NI-CER-2048CX-ADVPREM-AC, NI-CER-2048CX-ADVPREM-DC, NI-CER-2048F-ADVPREM-AC, NI-CER-2048F-ADVPREM-DC, NI-CER-2048FX-ADVPREM-AC, NI-CER-2048FX-ADVPREM-DC, BR-CER-2024C-4X-RT-AC, BR-CER-2024C-4X-RT-DC, BR-CER-2024F-4X-RT-AC, and BR-CER-2024F-4X-RT-DC); and
- Brocade NetIron CES 2000 Series Hardware Platforms (BR-CES-2024C-4X-AC, BR-CES-2024C-4X-DC, BR-CES-2024F-4X-AC, and BR-CES-2024F-4X-DC).

The following links offer additional information about each series of the TOE:

- **Brocade MLX Series**

  http://www.brocade.com/products/all/routers/product-details/netiron-mlx-series/index.page

  http://www.brocade.com/forms/getFile?p=documents/data_sheets/product_data_sheets/brocade-mlx-series-ds.pdf

- **Brocade NetIron CER 2000 Series**

  http://www.brocade.com/products/all/routers/product-details/netiron-cer-2000-series/index.page

  http://www.brocade.com/forms/getFile?p=documents/data_sheets/product_data_sheets/brocade-netiron-cer-2000-ds.pdf

- **Brocade NetIron CES 2000 Series**

http://www.brocade.com/products/all/switches/product-details/netiron-ces-2000-series/index.page

http://www.brocade.com/downloads/documents/data_sheets/product_data_sheets/brocade-netiron-ces-2000-ds.pdf

While there are different models in the TOE, they differ primarily in physical form factor, number and types of connections and slots, and relative performance. There are some functional differences among the families, but they each provide the same security characteristics as claimed in this security target.

The different series have differing CPUs as described below

- The MLX Series uses a Freescale MPC 7448, 1700 MHz CPU for the MR2 models and
- The CER 2000 and CES 2000 Series utilize a Freescale MPC8544, PowerQUICC[tm] 800 MHz CPU

1.4.1 TOE Architecture

The basic architecture of each TOE appliance begins with a hardware appliance with physical network connections. Within the hardware appliance, the Brocade IronWare OS is designed to control and enable access to the available hardware functions (e.g., program execution, device access, facilitate basic routing functions). IronWare OS enforces applicable security policies on network information flowing through the hardware appliance.

The basic start-up operation of the TOE is as follows:

1. At system startup the operating system is transferred from flash memory to dynamic memory using a built-in hardware bootstrap.
2. The operating system reads the configuration parameters from the configuration file in non-volatile memory and then builds the necessary data structures in dynamic memory and begins operation.

During normal operation, IP packets are sent to the management IP address or through the appliance over one or more of its physical network interfaces, which processes them according to the system's configuration and state information dynamically maintained by the appliance. This processing typically results in the frames or packets being forwarded out of the device over another interface. The TOE will process other packets destined for itself (control path packets) based on the requirements of the given protocol (HTTPS or SSH).

1.4.1.1 Physical Boundaries

Each TOE appliance runs a version of the Brocades software and has physical network connections to its environment to facilitate routing and switching of network traffic. The TOE appliance can also be the destination of network traffic, where it provides interfaces for its own management.

The TOE may be accessed and managed through a PC or terminal in the environment which can be remote from or directly connected to the TOE.

The TOE can be configured to forward its audit records to an external syslog server in the network environment. This is generally advisable given the limited audit log storage space on the evaluated appliances.

The TOE can be configured to synchronize its internal clock using an external NTP server in the operational environment.

NetIron (unlike the Broacde FastIron series, which provides no SSL encryption for external authentication servers) provides SSL encrypted TACACS+ authentication but does not provide SSL encrypted RADIUS. Thus, the use of RADIUS external authentication services are excluded from the evaluated configuration of the TOE. NetIron's TACACS+ supports password authentication only and does not support SSH public-ley authentication.

1.4.1.2 Logical Boundaries

This section summarizes the security functions provided by the Brocade MLXe® and NetIron® Family Devices with Multi-Service IronWare R05.8.00: The TOE logical boundary consists of the security functionality of the products summarized in the following subsections

- Security audit
- Cryptographic support
- User data protection
- Identification and authentication
- Security management
- Protection of the TSF
- TOE access
- Trusted path/channels

Note that use of the following features is limited in the evaluated TOE:

1. The use of SNMP has **not** been subject to evaluation. Note that SNMP can be used only to monitor as SNMP cannot access any security related parameters.
2. The *Strict Password Enforcement* setting is assumed to be **enabled** in the evaluated configuration.
3. The TOE will be operated in Common Criteria mode (a more restricted mode than FIPS mode).

Given that this Security Target conforms to the NDPP, the security claims focus on the TOE as a secure network infrastructure device and do not focus on other key functions provided by the TOE, such as controlling the flow of network packets among the attached networks. However, those functions can be freely used without affecting the claimed and evaluated security functions; they simply have not been evaluated to work correctly themselves.

The TOE protects itself from tampering and bypass by offering only a limited and controlled set of functions at each of its physical interfaces to its environment. Communication via those interfaces is either directed at the TOE for the purpose of administration or is directed through the TOE for communication among network devices. In both cases the TOE implements a set of policies to control the services available and those services are designed to protect and ensure the secure operation of the TOE.

The TOE includes the ability to communicate with a SYSLOG server in its environment to access its services. The TOE is designed to interact with each of those servers in accordance with their respective protocols, including security capabilities where applicable.

1.4.1.2.1 Security audit

The TOE is designed to be able to generate logs for a wide range of security relevant events. The TOE can be configured to store the logs locally so they can be accessed by an administrator and also to send the logs to a designated log server using TLS to protect the logs while in transit on the network.

1.4.1.2.2 Cryptographic support

The TOE is a FIPS-validated cryptographic module that provides key management, random bit generation, encryption/decryption, digital signature and secure hashing and key-hashing features in support of higher level cryptographic protocols including SSH and TLS/HTTPS.

1.4.1.2.3 User data protection

The TOE performs a wide variety of network switching and routing functions, passing network traffic among its various network connections. While implementing applicable network protocols associated with network traffic routing, the TOE is carefully designed to ensure that it doesn't inadvertently reuse data found in network traffic.

This is accomplished primarily by controlling the size of all buffers, fully overwriting buffer contents, and zero-padding of memory structures and buffers when necessary.

1.4.1.2.4 Identification and authentication

The TOE requires users to be identified and authenticated before they can use functions mediated by the TOE, with the exception of passing network traffic in accordance with its configured switching/routing rules. It provides the ability to both assign attributes (user names, passwords and privilege levels) and to authenticate users against these attributes.

1.4.1.2.5 Security management

The TOE provides Command Line Interface (CLI) commands and the MLX series provides an HTTPS (utilizing TLS v1.0) Graphical User Interface (Web GUI) to access the wide range of security management functions to manage its security policies. All administrative activity and functions including security management commands are limited to authorized users (i.e., administrators) only after they have provided acceptable user identification and authentication data to the TOE. The security management functions are controlled through the use of privileges associated with roles that can be assigned to TOE users. Among the available privileges, only the Super User can actually manage the security policies provided by the TOE and the TOE offers a complete set of functions to facilitate effective management since the Super User allows for complete read-and-write access to the system.

1.4.1.2.6 Protection of the TSF

The TOE implements a number of features design to protect itself to ensure the reliability and integrity of its security features.

It protects particularly sensitive data such as stored passwords and cryptographic keys so that they are not accessible even by an administrator. It also provides its own timing mechanism to ensure that reliable time information is available (e.g., for log accountability).

Note that the TOE is a single appliance, and as such, no intra-TOE communication is subject to any risks that may require special protection (e.g., cryptographic mechanisms).

The TOE includes functions to perform self-tests so that it might detect when it is failing. It also includes mechanisms (i.e., verification of the digital signature of each new image) so that the TOE itself can be updated while ensuring that the updates will not introduce malicious or other unexpected changes in the TOE.

1.4.1.2.7 TOE access

The TOE can be configured to display a message of the day banner when an administrator establishes an interactive session and subsequently will enforce an administrator-defined inactivity timeout value after which the inactive session (local or remote) will be terminated.

1.4.1.2.8 Trusted path/channels

The TOE protects interactive communication with administrators using SSHv2 for CLI access or, for the MLX series, TLS/HTTPS for Web graphical user interface access. In each case, the both integrity and disclosure protection is ensured. If the negotiation of an encrypted session fails or if the user does not have authorization for remote administration, the attempted connection will not be established.

The TOE protects communication with network peers, such as a log server, using TLS connections to prevent unintended disclosure or modification of logs.

1.4.2 TOE Documentation

Brocade offers a series of documents that describe the installation of the Brocade MLXe® and NetIron® Family Devices with Multi-Service IronWare R05.8.00 as well as guidance for subsequent use and administration of the applicable security features. The following list of documents was examined as part of the evaluation:

- Multi-Service IronWare Federal Information Processing Standards and Common Criteria Guide Supporting Multi-Service IronWare R05.8.00a, 53-1003269-01, 20 March 2015.
- Multi-Service IronWare Administration Configuration Guide Supporting Multi-Service IronWare R05.8.00, 53-1003254-01, 13 January 2015.
- Multi-Service IronWare Security Configuration Guide Supporting Multi-Service IronWare R05.8.00, 53-1003255-01, 13 January 2015.

2. Conformance Claims

This TOE is conformant to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, Version 3.1, Revision 4, September 2012.
    - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1 Revision 4, September 2102.
    - Part 3 Conformant
- The ST conforms to the *Protection Profile for Network Devices*, version 1.1, 8 June 2012 (NDPP) with Errata #3, 3 November 2014
- Package Claims:
    - Assurance Level: EAL 1 conformant

      The NDPP defines assurance activities beyond the scope for EAL 1, and this Security Target includes them to ensure that they are within scope of the corresponding evaluation. However, at the present time, international recognition of the evaluation results are limited to defined assurance packages, such as EAL1, and does not extend to Scheme-defined assurance extensions or refinements.

2.1 Conformance Rationale

The ST conforms to the *Protection Profile for Network Devices*, version 1.1, 8 June 2012 (NDPP) with Errata #3, 3 November 2014. As explained previously, the security problem definition, security objectives, and security requirements have been drawn from the NDPP.

3. Security Objectives

The Security Problem Definition may be found in the *Protection Profile for Network Devices*, version 1.1, 8 June 2012 (NDPP) with Errata #3, 3 November 2014, and this section reproduces only the corresponding Security Objectives for convenience. The NDPP offers additional information about the identified security objectives, but that has not been reproduced here and the NDPP should be consulted if there is interest in that material.

In general, the NDPP has presented a Security Objectives appropriate for network infrastructure devices and as such are applicable to the Brocade MLXe® and NetIron® Family Devices with Multi-Service IronWare R05.8.00.

3.1 Security Objectives for the TOE

O.DISPLAY_BANNER

The TOE will display an advisory warning regarding use of the TOE.

O.PROTECTED_COMMUNICATIONS

The TOE will provide protected communication channels for administrators, other parts of a distributed TOE, and authorized IT entities.

O.RESIDUAL_INFORMATION_CLEARING

The TOE will ensure that any data contained in a protected resource is not available when the resource is reallocated.

O.SESSION_LOCK

The TOE shall provide mechanisms that mitigate the risk of unattended sessions being hijacked.

O.SYSTEM_MONITORING

The TOE will provide the capability to generate audit data and send those data to an external IT entity.

O.TOE_ADMINISTRATION

The TOE will provide mechanisms to ensure that only administrators are able to log in and configure the TOE, and provide protections for logged-in administrators.

O.TSF_SELF_TEST

The TOE will provide the capability to test some subset of its security functionality to ensure it is operating properly.

O.VERIFIABLE_UPDATES

The TOE will provide the capability to help ensure that any updates to the TOE can be verified by the administrator to be unaltered and (optionally) from a trusted source.

3.2 Security Objectives for the Operational 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.PHYSICAL Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.

OE.TRUSTED_ADMIN
TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.

4. Extended Components Definition

All of the extended requirements in this ST have been drawn from the NDPP. The NDPP defines the following extended SFRs and since they are not redefined in this ST the NDPP should be consulted for more information in regard to those CC extensions.

- FAU_STG_EXT.1: External Audit Trail Storage
- FCS_CKM_EXT.4: Cryptographic Key Zeroization
- FCS_HTTPS_EXT.1: Explicit: HTTPS
- FCS_RBG_EXT.1: Extended: Cryptographic Operation (Random Bit Generation)
- FCS_SSH_EXT.1: Explicit: SSH
- FCS_TLS_EXT.1: Explicit: TLS
- FIA_PMG_EXT.1: Password Management
- FIA_UIA_EXT.1: User Identification and Authentication
- FIA_UAU_EXT.2: Extended: Password-based Authentication Mechanism
- FPT_APW_EXT.1: Extended: Protection of Administrator Passwords
- FPT_SKP_EXT.1: Extended: Protection of TSF Data (for reading of all symmetric keys)
- FPT_TST_EXT.1: TSF Testing
- FPT_TUD_EXT.1: Extended: Trusted Update
- FTA_SSL_EXT.1: TSF-initiated Session Locking

5. Security Requirements

This section defines the Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs) that serve to represent the security functional claims for the Target of Evaluation (TOE) and to scope the evaluation effort.

The SFRs have all been drawn from the Protection Profile (PP): *Protection Profile for Network Devices*, version 1.1, 8 June 2012 (NDPP) with Errata #3, 3 November 2014. The refinements and operations already performed in that PP are not identified (e.g., highlighted) here, rather the requirements have been copied from that PP and any residual operations have been completed herein. Of particular note, the NDPP made a number of refinements and completed some of the SFR operations defined in the Common Criteria (CC) and that PP should be consulted to identify those changes if necessary.

The SARs are also drawn from the NDPP which includes all the SARs for EAL1 as defined in the CC. However, the SARs are effectively refined since requirement-specific 'Assurance Activities' are defined in the NDPP that serve to ensure corresponding evaluations will yield more practical and consistent assurance than the EAL1 assurance requirements alone. As such, those assurance activities have been reproduced in this ST to ensure they are included within the scope of the evaluation effort.

5.1 TOE Security Functional Requirements

The following table identifies the SFRs that are satisfied by the TOE.

Requirement Class	Requirement Component
FAU: Security audit	FAU_GEN.1: Audit Data Generation
	FAU_GEN.2: User identity association
	FAU_STG_EXT.1: External Audit Trail Storage
FCS: Cryptographic support	FCS_CKM.1: Cryptographic Key Generation (for asymmetric keys)
	FCS_CKM_EXT.4: Cryptographic Key Zeroization
	FCS_COP.1(1): Cryptographic Operation (for data encryption/decryption)
	FCS_COP.1(2): Cryptographic Operation (for cryptographic signature)
	FCS_COP.1(3): Cryptographic Operation (for cryptographic hashing)
	FCS_COP.1(4): Cryptographic Operation (for keyed-hash message authentication)
	FCS_HTTPS_EXT.1: Explicit: HTTPS
	FCS_RBG_EXT.1: Extended: Cryptographic Operation (Random Bit Generation)
	FCS_SSH_EXT.1: Explicit: SSH
	FCS_TLS_EXT.1: Explicit: TLS
FDP: User data protection	FDP_RIP.2: Full Residual Information Protection
FIA: Identification and authentication	FIA_PMG_EXT.1: Password Management
	FIA_UAU.7: Protected Authentication Feedback
	FIA_UIA_EXT.1: User Identification and Authentication
	FIA_UAU_EXT.2: Extended: Password-based Authentication Mechanism
FMT: Security management	FMT_MTD.1: Management of TSF Data (for general TSF data)
	FMT_SMF.1: Specification of Management Functions
	FMT_SMR.2: Restrictions on Security Roles
FPT: Protection of the TSF	FPT_SKP_EXT.1: Extended: Protection of TSF Data (for reading of all symmetric keys)
	FPT_APW_EXT.1: Extended: Protection of Administrator Passwords
	FPT_STM.1: Reliable Time Stamps
	FPT_TST_EXT.1: TSF Testing
	FPT_TUD_EXT.1: Extended: Trusted Update

FTA: TOE access	FTA_SSL.3: TSF-initiated Termination
	FTA_SSL.4: User-initiated Termination
	FTA_SSL_EXT.1: TSF-initiated Session Locking
	FTA_TAB.1: Default TOE Access Banners
FTP: Trusted path/channels	FTP_ITC.1: Trusted Channel
	FTP_TRP.1: Trusted Path

Table 1 TOE Security Functional Components

5.1.1 Security Audit (FAU)

5.1.1.1 Audit Data Generation (FAU_GEN.1)

FAU_GEN.1.1

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

a) Start-up of the audit functions;
b) All auditable events for the not specified level of audit; and
c) All administrative actions;
d) Specifically defined auditable events listed in **Table 2 Auditable Events**.

Assurance Activity:

The evaluator shall check the administrative guide and ensure that it lists all of the auditable events and provides a format for audit records. Each audit record format type must be covered, along with a brief description of each field. The evaluator shall check to make sure that every audit event type mandated by the NDPP is described and that the description of the fields contains the information required in FAU_GEN.1.2, and the additional information specified in **Table 2 Auditable Events**.

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

The evaluator shall test the TOE's ability to correctly generate audit records by having the TOE generate audit records for the events listed in **Table 2 Auditable Events** and administrative actions. This should include all instances of an event--for instance, if there are several different I&A mechanisms for a system, the FIA_UIA_EXT.1 events must be generated for each mechanism. The evaluator shall test that audit records are generated for the establishment and termination of a channel for each of the cryptographic protocols contained in the ST. If HTTPS is implemented, the test demonstrating the establishment and termination of a TLS session can be combined with the test for an HTTPS session. For administrative actions, the evaluator shall test that each action determined by the evaluator above to be security relevant in the context of the NDPP is auditable. When verifying the test results, the evaluator shall ensure the audit records generated during testing match the format specified in the administrative guide, and that the fields in each audit record have the proper entries.

Note that the testing here can be accomplished in conjunction with the testing of the security mechanisms directly. For example, testing performed to ensure that the administrative guidance

provided is correct verifies that AGD_OPE.1 is satisfied and should address the invocation of the administrative actions that are needed to verify the audit records are generated as expected.

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 PP/ST, information specified in column three of **Table 2 Auditable Events**.

Assurance Activity:

This activity should be accomplished in conjunction with the testing of FAU_GEN.1.1.

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None.	
FAU_GEN.2	None.	
FAU_STG_EXT.1	None.	
FCS_CKM.1	None.	
FCS_CKM_EXT.4	None.	
FCS_COP.1(1)	None.	
FCS_COP.1(2)	None.	
FCS_COP.1(3)	None.	
FCS_COP.1(4)	None.	
FCS_HTTPS_EXT.1	Failure to establish a HTTPS Session. Establishment/Termination of a HTTPS session. [1]	Reason for failure. Non-TOE endpoint of connection (IP address) for both successes and failures.
FCS_RBG_EXT.1	None.	
FCS_SSH_EXT.1	Failure to establish an SSH session. Establishment/Termination of an SSH session. [1]	Reason for failure Non-TOE endpoint of connection (IP address) for both successes and failures.
FCS_TLS_EXT.1	Failure to establish a TLS Session. Establishment/Termination of a TLS session. [1]	Reason for failure. Non-TOE endpoint of connection (IP address) for both successes and failures.
FDP_RIP.2	None.	
FIA_PMG_EXT.1	None.	
FIA_UIA_EXT.1	All use of the identification and authentication mechanism.	Provided user identity, origin of the attempt (e.g., IP address).
FIA_UAU_EXT.2	All use of the authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UAU.7	None.	
FMT_MTD.1	None.	
FMT_SMF.1	None.	
FMT_SMR.2	None.	
FPT_SKP_EXT.1	None.	
FPT_APW_EXT.1	None.	
FPT_STM.1	Changes to the time.	The old and new values for the time. Origin of the attempt (e.g., IP address).

[1] Auditing session establishment failures is highly dependent on the implementation and is currently not standardized in the industry. In this ST, no specific list or types of such failures is mandated as being auditable. More specifically in this case, only user-level authentication failures are necessarily associated with SSH, HTTPS or TLS session establishment failure.

Requirement	Auditable Events	Additional Audit Record Contents
FPT_TUD_EXT.1	Initiation of update.	No additional information.
FPT_TST_EXT.1	None.	
FTA_SSL_EXT.1	Any attempts at unlocking of an interactive session.	No additional information.
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	No additional information.
FTA_SSL.4	The termination of an interactive session.	No additional information.
FTA_TAB.1	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	Initiation of the trusted channel. Termination of the trusted channel. Failures of the trusted path functions.	Identification of the claimed user identity.

Table 2 Auditable Events

5.1.1.2 User Identity Association (FAU_GEN.2)

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.

Assurance Activity:

This activity should be accomplished in conjunction with the testing of FAU_GEN.1.1.

5.1.1.3 External Audit Trail Storage (FAU_STG_EXT.1)

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 implementing the [***TLS***] protocol.

Assurance Activity:

The evaluator shall examine the TSS to ensure it describes the amount of audit data that are stored locally; what happens when the local audit data store is full; and how these records are protected against unauthorized access. The evaluator shall also examine the operational guidance to determine that it describes the relationship between the local audit data and the audit data that are sent to the audit log server (for TOEs that are not acting as an audit log server). For example, when an audit event is generated, is it simultaneously sent to the external server and the local store, or is the local store used as a buffer and "cleared" periodically by sending the data to the audit server.

The evaluator shall examine the TSS to ensure it describes the means by which the audit data are transferred to the external audit server, and how the trusted channel is provided. Testing of the trusted channel mechanism will be performed as specified in the associated assurance activities for the particular trusted channel mechanism. The evaluator shall also examine the operational guidance to ensure it describes how to establish the trusted channel to the audit server, as well as describe any requirements on the audit server (particular audit server protocol, version of the protocol required, etc.), as well as configuration of the TOE needed to communicate with the audit server. The evaluator shall perform the following test for this requirement:

Test 1: The evaluator shall establish a session between the TOE and the audit server according to the configuration guidance provided. The evaluator shall then examine the traffic that passes between the audit server and the TOE during several activities of the evaluator's choice designed to generate audit data to be transferred to the audit server. The evaluator shall observe that these data are not able to be viewed in the clear during this transfer, and that they are successfully received by the audit server. The evaluator shall record the particular software (name, version) used on the audit server during testing.

5.1.2 Cryptographic Support (FCS)

5.1.2.1 Cryptographic Key Generation (for asymmetric keys) (FCS_CKM.1)

FCS_CKM.1.1

Refinement: The TSF shall generate asymmetric cryptographic keys used for key establishment in accordance with **[**

- ***NIST Special Publication 800-56B, 'Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography' for RSA-based key establishment schemes*****]**

and specified cryptographic key sizes equivalent to, or greater than, a symmetric key strength of 112 bits.

Assurance Activity:

The evaluator shall use the key pair generation portions of "The FIPS 186-3 Digital Signature Algorithm Validation System (DSA2VS)", "The FIPS 186-3 Elliptic Curve Digital Signature Algorithm Validation System (ECDSA2VS)", and either "The RSA Validation System (RSAVS)" (for FIPS 186-2) or "The 186-3 RSA Validation System (RSA2VS)" (for FIPS 186-3) as a guide in testing the requirement above, depending on the selection performed by the ST author. This will require that the evaluator have a trusted reference implementation of the algorithms that can produce test vectors that are verifiable during the test.

The evaluator shall ensure that the TSS contains a description of how the TSF complies with 800-56A and/or 800-56B, depending on the selections made. This description shall indicate the sections in 800-56A and/or 800-56B that are implemented by the TSF, and the evaluator shall ensure that key establishment is among those sections that the TSF claims to implement.

Any TOE-specific extensions, processing that is not included in the documents, or alternative implementations allowed by the documents that may impact the security requirements the TOE is to enforce shall be described

5.1.2.2 Cryptographic Key Zeroization (FCS_CKM_EXT.4)

FCS_CKM_EXT.4.1

The TSF shall zeroize all plaintext secret and private cryptographic keys and CSPs when no longer required.

Assurance Activity:

The evaluator shall check to ensure the TSS describes each of the secret keys (keys used for symmetric encryption), private keys, and CSPs used to generate keys; when they are zeroized (for example, immediately after use, on system shutdown, etc.); and the type of zeroization procedure that is performed (overwrite with zeros, overwrite three times with random pattern, etc.). If different types of memory are used to store the materials to be protected, the evaluator shall check to ensure that the TSS describes the zeroization procedure in terms of the memory in which the data are stored (for example, "secret keys stored on flash are zeroized by overwriting once with

zeros, while secret keys stored on the internal hard drive are zeroized by overwriting three times with a random pattern that is changed before each write").

5.1.2.3 Cryptographic Operation (for data encryption/decryption) (FCS_COP.1(1))

FCS_COP.1(1).1

Refinement: The TSF shall perform encryption and decryption in accordance with a specified cryptographic algorithm AES operating in [**CBC**] and cryptographic key sizes 128-bits and 256-bits that meets the following:

- FIPS PUB 197, 'Advanced Encryption Standard (AES)' [***NIST SP 800-38A***]

Assurance Activity:

The evaluator shall use tests appropriate to the modes selected in the above requirement from "The Advanced Encryption Standard Algorithm Validation Suite (AESAVS)", "The XTS-AES Validation System (XTSVS)", The CMAC Validation System (CMACVS)", "The Counter with Cipher Block Chaining-Message Authentication Code (CCM) Validation System (CCMVS)", and "The Galois/Counter Mode (GCM) and GMAC Validation System (GCMVS)" (these documents are available from http://csrc.nist.gov/groups/STM/cavp/index.html) as a guide in testing the requirement above. This will require that the evaluator have a reference implementation of the algorithms known to be good that can produce test vectors that are verifiable during the test.

5.1.2.4 Cryptographic Operation (for cryptographic signature) (FCS_COP.1(2))

FCS_COP.1(2).1

Refinement: The TSF shall perform cryptographic signature services in accordance with a [

(1) RSA Digital Signature Algorithm (rDSA) with a key size (modulus) of 2048 bits or greater]

that meets the following:

Case: RSA Digital Signature Algorithm

- **FIPS PUB 186-2 or FIPS 186-3, 'Digital Signature Standard'.**

Assurance Activity:

The evaluator shall use the signature generation and signature verification portions of "The Digital Signature Algorithm Validation System" (DSA2VS), "The Elliptic Curve Digital Signature Algorithm Validation System" (ECDSA2VS), and "The RSA Validation System" (RSAVS (for 186-2) or RSA2VS (for 186-3)) as a guide in testing the requirement above. The Validation System used shall comply with the conformance standard identified in the ST (i.e., FIPS PUB 186-2 or FIPS PUB 186-3). This will require that the evaluator have a reference implementation of the algorithms known to be good that can produce test vectors that are verifiable during the test.

5.1.2.5 Cryptographic Operation (for cryptographic hashing) (FCS_COP.1(3))

FCS_COP.1(3).1

Refinement: The TSF shall perform cryptographic hashing services in accordance with a specified cryptographic algorithm [***SHA-1, SHA-224, SHA-256, SHA-384, SHA-512***] and message digest sizes [***160, 224, 256, 384, 512***] bits that meet the following: FIPS Pub 180-3, 'Secure Hash Standard.'

Assurance Activity:

The evaluator shall use "The Secure Hash Algorithm Validation System (SHAVS)" as a guide in testing the requirement above. This will require that the evaluator have a reference implementation of the algorithms known to be good that can produce test vectors that are verifiable during the test.

5.1.2.6 Cryptographic Operation (for keyed-hash message authentication) (FCS_COP.1(4))

FCS_COP.1(4).1

Refinement: The TSF shall perform keyed-hash message authentication in accordance with a specified cryptographic algorithm HMAC-[***SHA-1, SHA-224, SHA-256, SHA-384, SHA-512***], key size [**equal to the input block size**], and message digest sizes [***160***] bits that meet the following: FIPS Pub 198-1, 'The Keyed-Hash Message Authentication Code', and FIPS Pub 180-3, 'Secure Hash Standard.'

Assurance Activity:

The evaluator shall use "The Keyed-Hash Message Authentication Code (HMAC) Validation System (HMACVS)" as a guide in testing the requirement above. This will require that the evaluator have a reference implementation of the algorithms known to be good that can produce test vectors that are verifiable during the test.

5.1.2.7 Explicit: HTTPS (FCS_HTTPS_EXT.1)

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 as specified in FCS_TLS_EXT.1.

Component Assurance Activity:

The evaluator shall check the TSS to ensure that it is clear on how HTTPS uses TLS to establish an administrative session, focusing on any client authentication required by the TLS protocol vs. security administrator authentication which may be done at a different level of the processing stack. Testing for this activity is done as part of the TLS testing; this may result in additional testing if the TLS tests are done at the TLS protocol level.

5.1.2.8 Extended: Cryptographic Operation (Random Bit Generation) (FCS_RBG_EXT.1)

FCS_RBG_EXT.1.1

The TSF shall perform all random bit generation (RBG) services in accordance with [***NIST Special Publication 800-90 using [CTR_DRBG (AES-256)]***] seeded by an entropy source that accumulated entropy from [***a software-based noise source and a TSF-hardware-based noise source]***.

FCS_RBG_EXT.1.2

The deterministic RBG shall be seeded with a minimum of [***256 bits***] of entropy at least equal to the greatest security strength of the keys and hashes that it will generate.

Assurance Activity:

Documentation shall be produced—and the evaluator shall perform the activities—in accordance with Annex D, Entropy Documentation and Assessment.

> **Annex D: Entropy Documentation and Assessment**
>
> The documentation of the entropy source should be detailed enough that, after reading, the evaluator will thoroughly understand the entropy source and why it can be relied upon to provide entropy. This documentation should include multiple detailed sections: design description, entropy justification, operating conditions, and health testing. This documentation is not required to be part of the TSS.

Design Description

Documentation shall include the design of the entropy source as a whole, including the interaction of all entropy source components. It will describe the operation of the entropy source to include how it works, how entropy is produced, and how unprocessed (raw) data can be obtained from within the entropy source for testing purposes. The documentation should walk through the entropy source design indicating where the random comes from, where it is passed next, any post-processing of the raw outputs (hash, XOR, etc.), if/where it is stored, and finally, how it is output from the entropy source. Any conditions placed on the process (e.g., blocking) should also be described in the entropy source design. Diagrams and examples are encouraged.

This design must also include a description of the content of the security boundary of the entropy source and a description of how the security boundary ensures that an adversary outside the boundary cannot affect the entropy rate.

Entropy Justification

There should be a technical argument for where the unpredictability in the source comes from and why there is confidence in the entropy source exhibiting probabilistic behavior (an explanation of the probability distribution and justification for that distribution given the particular source is one way to describe this). This argument will include a description of the expected entropy rate and explain how you ensure that sufficient entropy is going into the TOE randomizer seeding process. This discussion will be part of a justification for why the entropy source can be relied upon to produce bits with entropy.

Operating Conditions

Documentation will also include the range of operating conditions under which the entropy source is expected to generate random data. It will clearly describe the measures that have been taken in the system design to ensure the entropy source continues to operate under those conditions. Similarly, documentation shall describe the conditions under which the entropy source is known to malfunction or become inconsistent. Methods used to detect failure or degradation of the source shall be included.

Health Testing
More specifically, all entropy source health tests and their rationale will be documented. This will include a description of the health tests, the rate and conditions under which each health test is performed (e.g., at startup, continuously, or on-demand), the expected results for each health test, and rationale indicating why each test is believed to be appropriate for detecting one or more failures in the entropy source.

The evaluator shall also perform the following tests, depending on the standard to which the RBG conforms.

Implementations Conforming to FIPS 140-2, Annex C

The reference for the tests contained in this section is The Random Number Generator Validation System (RNGVS) [RNGVS]. The evaluator shall conduct the following two tests. Note that the

'expected values' are produced by a reference implementation of the algorithm that is known to be correct. Proof of correctness is left to each Scheme.

The evaluator shall perform a Variable Seed Test. The evaluator shall provide a set of 128 (Seed, DT) pairs to the TSF RBG function, each 128 bits. The evaluator shall also provide a key (of the length appropriate to the AES algorithm) that is constant for all 128 (Seed, DT) pairs. The DT value is incremented by 1 for each set. The seed values shall have no repeats within the set. The evaluator ensures that the values returned by the TSF match the expected values.

The evaluator shall perform a Monte Carlo Test. For this test, they supply an initial Seed and DT value to the TSF RBG function; each of these is 128 bits. The evaluator shall also provide a key (of the length appropriate to the AES algorithm) that is constant throughout the test. The evaluator then invokes the TSF RBG 10,000 times, with the DT value being incremented by 1 on each iteration, and the new seed for the subsequent iteration produced as specified in NIST-Recommended Random Number Generator Based on ANSI X9.31 Appendix A.2.4 Using the 3-Key Triple DES and AES Algorithms, Section 3. The evaluator ensures that the 10,000th value produced matches the expected value.

Implementations Conforming to NIST Special Publication 800-90

The evaluator shall perform 15 trials for the RBG implementation. If the RBG is configurable, the evaluator shall perform 15 trials for each configuration. The evaluator shall also confirm that the operational guidance contains appropriate instructions for configuring the RBG functionality.

If the RBG has prediction resistance enabled, each trial consists of (1) instantiate drbg, (2) generate the first block of random bits (3) generate a second block of random bits (4) uninstantiate. The evaluator verifies that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 – 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The next two are additional input and entropy input for the first call to generate. The final two are additional input and entropy input for the second call to generate. These values are randomly generated. “generate one block of random bits” means to generate random bits with number of returned bits equal to the Output Block Length (as defined in NIST SP 800-90).

If the RBG does not have prediction resistance, each trial consists of (1) instantiate drbg, (2) generate the first block of random bits (3) reseed, (4) generate a second block of random bits (5) uninstantiate. The evaluator verifies that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 – 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The fifth value is additional input to the first call to generate. The sixth and seventh are additional input and entropy input to the call to reseed. The final value is additional input to the second generate call.

The following paragraphs contain more information on some of the input values to be generated/selected by the evaluator.

Entropy input: the length of the entropy input value must equal the seed length.

Nonce: If a nonce is supported (CTR_DRBG with no df does not use a nonce), the nonce bit length is one-half the seed length.

Personalization string: The length of the personalization string must be <= seed length. If the implementation only supports one personalization string length, then the same length can be used for both values. If more than one string length is support, the evaluator shall use personalization strings of two different lengths. If the implementation does not use a personalization string, no value needs to be supplied.

Additional input: the additional input bit lengths have the same defaults and restrictions as the personalization string lengths.

5.1.2.9 Explicit: SSH (FCS_SSH_EXT.1)

FCS_SSH_EXT.1.1

The TSF shall implement the SSH protocol that complies with RFCs 4251, 4252, 4253, 4254, and [***no other RFCs***].

FCS_SSH_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.

Assurance Activity:

The evaluator shall check to ensure that the TSS contains a description of the public key algorithms that are acceptable for use for authentication, that this list conforms to FCS_SSH_EXT.1.5, and ensure that password-based authentication methods are also allowed. The evaluator shall also perform the following tests:

Test 1: The evaluator shall, for each public key algorithm supported, show that the TOE supports the use of that public key algorithm to authenticate a user connection. Any configuration activities required to support this test shall be performed according to instructions in the operational guidance.

Test 2: Using the operational guidance, the evaluator shall configure the TOE to accept password-based authentication, and demonstrate that a user can be successfully authenticated to the TOE over SSH using a password as an authenticator.

FCS_SSH_EXT.1.3

The TSF shall ensure that, as described in RFC 4253, packets greater than [**256K**] bytes in an SSH transport connection are dropped.

Assurance Activity:

The evaluator shall check that the TSS describes how 'large packets' in terms of RFC 4253 are detected and handled. The evaluator shall also perform the following test:

Test 1: The evaluator shall demonstrate that if the TOE receives a packet larger than that specified in this component, that packet is dropped.

FCS_SSH_EXT.1.4

The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms: AES-CBC-128, AES-CBC-256, [***no other algorithms***].

Assurance Activity:

The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that optional characteristics are specified, and the encryption algorithms supported are specified as well. The evaluator shall check the TSS to ensure that the encryption algorithms specified are identical to those listed for this component. The evaluator shall also check the operational guidance to ensure that it contains instructions on configuring the TOE so that SSH conforms to the description in the TSS (for instance, the set of algorithms advertised by the TOE may have to be restricted to meet the requirements). The evaluator shall also perform the following test:

Test 1: The evaluator shall establish a SSH connection using each of the encryption

algorithms specified by the requirement. It is sufficient to observe (on the wire) the successful negotiation of the algorithm to satisfy the intent of the test.

FCS_SSH_EXT.1.5

The TSF shall ensure that the SSH transport implementation uses SSH_RSA and [***no other public key algorithms***] as its public key algorithm(s).

Assurance Activity:

The assurance activity associated with FCS_SSH_EXT.1.4 verifies this requirement.

FCS_SSH_EXT.1.6

The TSF shall ensure that data integrity algorithms used in SSH transport connection is [***hmac-sha1***].

Assurance Activity:

The evaluator shall check the TSS to ensure that it lists the supported data integrity algorithms, and that that list corresponds to the list in this component. The evaluator shall also check the operational guidance to ensure that it contains instructions to the administrator on how to ensure that only the allowed data integrity algorithms are used in SSH connections with the TOE (specifically, that the 'none' MAC algorithm is not allowed). The evaluator shall also perform the following test:

Test 1: The evaluator shall establish a SSH connection using each of the integrity algorithms specified by the requirement. It is sufficient to observe (on the wire) the successful negotiation of the algorithm to satisfy the intent of the test.

FCS_SSH_EXT.1.7

The TSF shall ensure that diffie-hellman-group14-sha1 and [***no other methods***] are the only allowed key exchange method used for the SSH protocol.

Assurance Activity:

The evaluator shall ensure that operational guidance contains configuration information that will allow the security administrator to configure the TOE so that all key exchanges for SSH are performed using DH group 14 and any groups specified from the selection in the ST. If this capability is 'hard-coded' into the TOE, the evaluator shall check the TSS to ensure that this is stated in the discussion of the SSH protocol. The evaluator shall also perform the following test:

Test 1: The evaluator shall attempt to perform a diffie-hellman-group1-sha1 key exchange, and observe that the attempt fails. . For each allowed key exchange method, the evaluator shall then attempt to perform a key exchange using that method, and observe that the attempt succeeds.

5.1.2.10 Explicit: TLS (FCS_TLS_EXT.1)

FCS_TLS_EXT.1.1

The TSF shall implement one or more of the following protocols [***TLS 1.0 (RFC 2246), TLS 1.1 (RFC 4346), TLS 1.2 (RFC 5246)***] supporting the following ciphersuites:

Mandatory Ciphersuites:

TLS_RSA_WITH_AES_128_CBC_SHA,

Optional Ciphersuites:

[***TLS_RSA_WITH_AES_256_CBC_SHA,***
TLS_DHE_RSA_WITH_AES_128_CBC_SHA,
TLS_DHE_RSA_WITH_AES_256_CBC_SHA]

Assurance Activity:

The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that the ciphersuites supported are specified. The evaluator shall check the TSS to ensure that the ciphersuites specified are identical to those listed for this component. The evaluator shall also check the operational guidance to ensure that it contains instructions on configuring the TOE so that TLS conforms to the description in the TSS (for instance, the set of ciphersuites advertised by the TOE may have to be restricted to meet the requirements). The evaluator shall also perform the following test:

- Test 1: The evaluator shall establish a TLS connection using each of the ciphersuites specified by the requirement. This connection may be established as part of the establishment of a higher-level protocol, e.g., as part of a HTTPS session. It is sufficient to observe the successful negotiation of a ciphersuite to satisfy the intent of the test; it is not necessary to examine the characteristics of the encrypted traffic in an attempt to discern the ciphersuite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES).

5.1.3 User Data Protection (FDP)

5.1.3.1 Full Residual Information Protection (FDP_RIP.2)

FDP_RIP.2.1

The TSF shall ensure that any previous information content of a resource is made unavailable upon the [***allocation of the resource to***] all objects.

Assurance Activity:

"Resources" in the context of this requirement are network packets being sent through (as opposed to "to", as is the case when a security administrator connects to the TOE) the TOE. The concern is that once a network packet is sent, the buffer or memory area used by the packet still contains data from that packet, and that if that buffer is re-used, those data might remain and make their way into a new packet. The evaluator shall check to ensure that the TSS describes packet processing to the extent that they can determine that no data will be reused when processing network packets. The evaluator shall ensure that this description at a minimum describes how the previous data are zeroized/overwritten, and at what point in the buffer processing this occurs.

5.1.4 Identification and Authentication (FIA)

5.1.4.1 Password Management (FIA_PMG_EXT.1)

FIA_PMG_EXT.1.1

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

1. Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: [***"!", "@", "#", "$", "%", "^", "&", "*", "(", ")", [" ' ", "+", ",", "-", ".", "/", ":", ";", "<", "=", ">", "[", "\", "]", "_", "`", "{", "}", and "~"]***];
2. Minimum password length shall settable by the Security Administrator, and support passwords of 15 characters or greater.

Assurance Activity:

The evaluator shall examine the operational guidance to determine that it provides guidance to security administrators on the composition of strong passwords, and that it provides instructions

on setting the minimum password length. The evaluator shall also perform the following tests. Note that one or more of these tests can be performed with a single test case.

Test 1: The evaluator shall compose passwords that either meet the requirements, or fail to meet the requirements, in some way. For each password, the evaluator shall verify that the TOE supports the password. While the evaluator is not required (nor is it feasible) to test all possible compositions of passwords, the evaluator shall ensure that all characters, rule characteristics, and a minimum length listed in the requirement are supported, and justify the subset of those characters chosen for testing.

5.1.4.2 Protected Authentication Feedback (FIA_UAU.7)

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.

Assurance Activity:

The evaluator shall perform the following test for each method of local login allowed:

Test 1: The evaluator shall locally authenticate to the TOE. While making this attempt, the evaluator shall verify that at most obscured feedback is provided while entering the authentication information.

5.1.4.3 Extended: Password-based Authentication Mechanism (FIA_UAU_EXT.2)

FIA_UAU_EXT.2.1

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

Component Assurance Activity:

Assurance activities for this requirement are covered under those for FIA_UIA_EXT.1. If other authentication mechanisms are specified, the evaluator shall include those methods in the activities for FIA_UIA_EXT.1.

5.1.4.4 User Identification and Authentication (FIA_UIA_EXT.1)

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;
- [***[network routing services]***].

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.

Component Assurance Activity:

The evaluator shall examine the TSS to determine that it describes the logon process for each logon method (local, remote (HTTPS, SSH, etc.)) supported for the product. This description shall contain information pertaining to the credentials allowed/used, any protocol transactions that take place, and what constitutes a "successful logon". The evaluator shall examine the operational guidance to determine that any necessary preparatory steps (e.g., establishing credential material such as pre-shared keys, tunnels, certificates, etc.) to logging in are described. For each supported the login method, the evaluator shall ensure the operational guidance provides clear instructions for successfully logging on. If configuration is necessary to ensure the services provided before

login are limited, the evaluator shall determine that the operational guidance provides sufficient instruction on limiting the allowed services.

The evaluator shall perform the following tests for each method by which administrators access the TOE (local and remote), as well as for each type of credential supported by the login method:

Test 1: The evaluator shall use the operational guidance to configure the appropriate credential supported for the login method. For that credential/login method, the evaluator shall show that providing correct I&A information results in the ability to access the system, while providing incorrect information results in denial of access.

Test 2: The evaluator shall configure the services allowed (if any) according to the operational guidance, and then determine the services available to an external remote entity. The evaluator shall determine that the list of services available is limited to those specified in the requirement.

Test 3: For local access, the evaluator shall determine what services are available to a local administrator prior to logging in, and make sure this list is consistent with the requirement.

5.1.5 Security management (FMT)

5.1.5.1 Management of TSF Data (for general TSF data) (FMT_MTD.1)

FMT_MTD.1.1

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

Assurance Activity:

The evaluator shall review the operational guidance to determine that each of the TSF-data-manipulating functions implemented in response to the requirements of the NDPP is identified, and that configuration information is provided to ensure that only administrators have access to the functions. The evaluator shall examine the TSS to determine that, for each administrative function identified in the operational guidance; those that are accessible through an interface prior to administrator log-in are identified. For each of these functions, the evaluator shall also confirm that the TSS details how the ability to manipulate the TSF data through these interfaces is disallowed for non-administrative users.

5.1.5.2 Specification of Management Functions (FMT_SMF.1)

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 update the TOE, and to verify the updates using [***digital signature***] capability prior to installing those updates; **[**
- ***Ability to configure the list of TOE-provided services available before an entity is identified and authenticated, as specified in FIA_UIA_EXT.1;***
- ***Ability to configure the cryptographic functionality*****]**.

Component Assurance Activity:

The security management functions for FMT_SMF.1 are distributed throughout the NDPP and are included as part of the requirements in FMT_MTD, FPT_TST_EXT, and any cryptographic management functions specified in the reference standards. Compliance to these requirements satisfies compliance with FMT_SMF.1.

5.1.5.3 Restrictions on Security Roles (FMT_SMR.2)

FMT_SMR.2.1

The TSF shall maintain the roles: Authorized 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

o Authorized Administrator role shall be able to administer the TOE locally;

o Authorized Administrator role shall be able to administer the TOE remotely;

are satisfied.

Component Assurance Activity:

The evaluator shall review the operational guidance to ensure that it contains instructions for administering the TOE both locally and remotely, including any configuration that needs to be performed on the client for remote administration. In the course of performing the testing activities for the evaluation, the evaluator shall use all supported interfaces, although it is not necessary to repeat each test involving an administrative action with each interface. The evaluator shall ensure, however, that each supported method of administering the TOE that conforms to the requirements of the NDPP be tested; for instance, if the TOE can be administered through a local hardware interface; SSH; and TLS/HTTPS; then all three methods of administration must be exercised during the evaluation team's test activities.

5.1.6 Protection of the TSF (FPT)

5.1.6.1 Extended: Protection of Administrator Passwords (FPT_APW_EXT.1)

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.

Component Assurance Activity:

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

5.1.6.2 Extended: Protection of TSF Data (for reading of all symmetric keys) (FPT_SKP_EXT.1)

FPT_SKP_EXT.1.1

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

Assurance Activity:

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

5.1.6.3 Reliable Time Stamps (FPT_STM.1)

FPT_STM.1.1

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

Assurance Activity:

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

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

Test 1: The evaluator uses the operational guide to set the time. The evaluator shall then use an available interface to observe that the time was set correctly.

Test2: [conditional] If the TOE supports the use of an NTP server; the evaluator shall use the operational guidance to configure the NTP client on the TOE, and set up a communication path with the NTP server. The evaluator will observe that the NTP server has set the time to what is expected. If the TOE supports multiple protocols for establishing a connection with the NTP server, the evaluator shall perform this test using each supported protocol claimed in the operational guidance.

5.1.6.4 TSF Testing (FPT_TST_EXT.1)

FPT_TST_EXT.1.1

The TSF shall run a suite of self-tests during initial start-up (on power on) to demonstrate the correct operation of the TSF.

Assurance Activity:

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

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

5.1.6.5 Extended: Trusted Update (FPT_TUD_EXT.1)

FPT_TUD_EXT.1.1

The TSF shall provide security administrators the ability to query the current version of the TOE firmware/software.

FPT_TUD_EXT.1.2

The TSF shall provide security administrators the ability to initiate updates to TOE firmware/software.

FPT_TUD_EXT.1.3

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

Component Assurance Activity:

Updates to the TOE either have a hash associated with them, or are signed by an authorized source. If digital signatures are used, the definition of an authorized source is contained in the TSS, along with a description of how the certificates used by the update verification mechanism are contained on the device. The evaluator ensures this information is contained in the TSS. The evaluator also ensures that the TSS (or the operational guidance) describes how the candidate updates are obtained; the processing associated with verifying the digital signature or calculating the hash of the updates; and the actions that take place for successful (hash or signature was verified) and unsuccessful (hash or signature could not be verified) cases. The evaluator shall perform the following tests:

Test 1: The evaluator performs the version verification activity to determine the current version of the product. The evaluator obtains a legitimate update using procedures described in the operational guidance and verifies that it is successfully installed on the TOE. Then, the evaluator performs a subset of other assurance activity tests to demonstrate that the update functions as expected. After the update, the evaluator performs the version verification activity again to verify the version correctly corresponds to that of the update.

Test 2: The evaluator performs the version verification activity to determine the current version of the product. The evaluator obtains or produces an illegitimate update, and attempts to install it on the TOE. The evaluator verifies that the TOE rejects the update.

5.1.7 TOE access (FTA)

5.1.7.1 TSF-initiated Termination (FTA_SSL.3)

FTA_SSL.3.1

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

Assurance Activity:

The evaluator shall perform the following test:

Test 1: The evaluator follows the operational guidance to configure several different values for the inactivity time period referenced in the component. For each period configured, the evaluator establishes a remote interactive session with the TOE. The evaluator then observes that the session is terminated after the configured time period.

5.1.7.2 User-initiated Termination (FTA_SSL.4)

FTA_SSL.4.1

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

Assurance Activity:

The evaluator shall perform the following test:

Test 1: The evaluator initiates an interactive local session with the TOE. The evaluator then follows the operational guidance to exit or log off the session and observes that the session has been terminated.

Test 2: The evaluator initiates an interactive remote session with the TOE. The evaluator then follows the operational guidance to exit or log off the session and observes that the session has been terminated.

5.1.7.3 TSF-initiated Session Locking (FTA_SSL_EXT.1)

FTA_SSL_EXT.1.1

The TSF shall, for local interactive sessions, [***terminate the session***] after a Security Administrator-specified time period of inactivity.

Assurance Activity:

The evaluator shall perform the following test:

Test 1: The evaluator follows the operational guidance to configure several different values for the inactivity time period referenced in the component. For each period configured, the evaluator establishes a local interactive session with the TOE. The evaluator then observes that the session is either locked or terminated after the configured time period. If locking was selected from the component, the evaluator then ensures that re-authentication is needed when trying to unlock the session.

5.1.7.4 Default TOE Access Banners (FTA_TAB.1)

FTA_TAB.1.1

Refinement: 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.

Assurance Activity:

The evaluator shall check the TSS to ensure that it details each method of access (local and remote) available to the administrator (e.g., serial port, SSH, HTTPS). The evaluator shall also perform the following test:

Test 1: The evaluator follows the operational guidance to configure a notice and consent warning message. The evaluator shall then, for each method of access specified in the TSS, establish a session with the TOE. The evaluator shall verify that the notice and consent warning message is displayed in each instance.

5.1.8 Trusted path/channels (FTP)

5.1.8.1 Trusted Channel (FTP_ITC.1)

FTP_ITC.1.1

Refinement: The TSF shall use [***TLS, SSH***] to provide a trusted communication channel between itself and authorized IT entities supporting the following capabilities: audit server, [***/TOE update server/***] 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 [**transmitting audit records to an audit server, retrieving a firmware update**].

Component Assurance Activity:

The evaluator shall examine the TSS to determine that, for all communications with authorized IT entities identified in the requirement, each communications mechanism is identified in terms of the allowed protocols for that IT entity. The evaluator shall also confirm that all protocols listed in the TSS are specified and included in the requirements in the ST. The evaluator shall confirm that the operational guidance contains instructions for establishing the allowed protocols with each authorized IT entity, and that it contains recovery instructions should a connection be unintentionally broken. The evaluator shall also perform the following tests:

Test 1: The evaluators shall ensure that communications using each protocol with each authorized IT entity is tested during the course of the evaluation, setting up the connections as described in the operational guidance and ensuring that communication is successful.

Test 2: For each protocol that the TOE can initiate as defined in the requirement, the evaluator shall follow the operational guidance to ensure that in fact the communication channel can be initiated from the TOE.

Test 3: The evaluator shall ensure, for each communication channel with an authorized IT entity, the channel data are not sent in plaintext.

Test 4: The evaluators shall, for each protocol associated with each authorized IT entity tested during test 1, the connection is physically interrupted. The evaluator shall ensure that when physical connectivity is restored, communications are appropriately protected.

Further assurance activities are associated with the specific protocols.

5.1.8.2 Trusted Path (FTP_TRP.1)

FTP_TRP.1.1

Refinement: The TSF shall use [***SSH or TLS/HTTPS***] **to** provide a trusted communication path between itself and remote administrators that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from disclosure and detection of modification of the communicated data.

FTP_TRP.1.2

Refinement: The TSF shall permit remote administrators to initiate communication via the trusted path.

FTP_TRP.1.3

The TSF shall require the use of the trusted path for initial administrator authentication and all remote administrative actions.

Component Assurance Activity:

The evaluator shall examine the TSS to determine that the methods of remote TOE administration are indicated, along with how those communications are protected. The evaluator shall also confirm that all protocols listed in the TSS in support of TOE administration are consistent with those specified in the requirement, and are included in the requirements in the ST. The evaluator shall confirm that the operational guidance contains instructions for establishing the remote administrative sessions for each supported method. The evaluator shall also perform the following tests:

Test 1: The evaluators shall ensure that communications using each specified (in the operational guidance) remote administration method is tested during the course of the evaluation, setting up the connections as described in the operational guidance and ensuring that communication is successful.

Test 2: For each method of remote administration supported, the evaluator shall follow the operational guidance to ensure that there is no available interface that can be used by a remote user to establish a remote administrative sessions without invoking the trusted path.

Test 3: The evaluator shall ensure, for each method of remote administration, the channel data are not sent in plaintext.

Further assurance activities are associated with the specific protocols.

5.2 TOE Security Assurance Requirements

The SARs for the TOE are the EAL 1 components as specified in Part 3 of the Common Criteria. Note that the SARs have effectively been refined with the assurance activities explicitly defined in association with both the SFRs and SARs.

Requirement Class	Requirement Component
ADV: Development	ADV_FSP.1: Basic functional specification
AGD: Guidance documents	AGD_OPE.1: Operational user guidance
	AGD_PRE.1: Preparative procedures
ALC: Life-cycle support	ALC_CMC.1: Labelling of the TOE
	ALC_CMS.1: TOE CM coverage
ATE: Tests	ATE_IND.1: Independent testing - conformance
AVA: Vulnerability assessment	AVA_VAN.1: Vulnerability survey

Table 3 EAL 1 Assurance Components

5.2.1 Development (ADV)

5.2.1.1 Basic Functional Specification (ADV_FSP.1)

ADV_FSP.1.1d

The developer shall provide a functional specification.

ADV_FSP.1.2d

The developer shall provide a tracing from the functional specification to the SFRs.

ADV_FSP.1.1c

The functional specification shall describe the purpose and method of use for each SFR-enforcing and SFR-supporting TSFI.

ADV_FSP.1.2c

The functional specification shall identify all parameters associated with each SFR-enforcing and SFR-supporting TSFI.

ADV_FSP.1.3c

The functional specification shall provide rationale for the implicit categorisation of interfaces as SFR-non-interfering.

ADV_FSP.1.4c

The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

ADV_FSP.1.1e

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_FSP.1.2e

The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

Component Assurance Activity:

There are no specific assurance activities associated with these SARs. The functional specification documentation is provided to support the evaluation activities described in Section 4.2 **of the NDPP**, and other activities described for AGD, ATE, and AVA SARs. The requirements on the content of the functional specification information is implicitly assessed by virtue of the other assurance activities being performed; if the evaluator is unable to perform an activity because the there is insufficient interface information, then an adequate functional specification has not been provided.

5.2.2 Guidance documents (AGD)

5.2.2.1 Operational user guidance (AGD_OPE.1)

AGD_OPE.1.1d

The developer shall provide operational user guidance.

AGD_OPE.1.1c

The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.

AGD_OPE.1.2c

The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.

AGD_OPE.1.3c

The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.

AGD_OPE.1.4c

The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_OPE.1.5c

The operational user guidance shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AGD_OPE.1.6c

The operational user guidance shall, for each user role, describe the security measures to be followed in order to fulfill the security objectives for the operational environment as described in the ST.

AGD_OPE.1.7c

The operational user guidance shall be clear and reasonable.

AGD_OPE.1.1e

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

Component Assurance Activity:

Some of the contents of the operational guidance will be verified by the assurance activities in Section 4.2 **of the NDPP** and evaluation of the TOE according to the CEM. The following additional information is also required.

The operational guidance shall at a minimum list the processes running (or that could run) on the TOE in its evaluated configuration during its operation that are capable of processing data received on the network interfaces (there are likely more than one of these, and this is not limited

to the process that 'listens' on the network interface). It is acceptable to list all processes running (or that could run) on the TOE in its evaluated configuration instead of attempting to determine just those that process the network data. For each process listed, the administrative guidance will contain a short (e.g., one- or two-line) description of the process' function, and the privilege with which the service runs. 'Privilege' includes the hardware privilege level (e.g., ring 0, ring 1), any software privileges specifically associated with the process, and the privileges associated with the user role the process runs as or under.

The operational guidance shall contain instructions for configuring the cryptographic engine associated with the evaluated configuration of the TOE. It shall provide a warning to the administrator that use of other cryptographic engines was not evaluated nor tested during the CC evaluation of the TOE.

The documentation must describe the process for verifying updates to the TOE, either by checking the hash or by verifying a digital signature. The evaluator shall verify that this process includes the following steps:

1. For hashes, a description of where the hash for a given update can be obtained. For digital signatures, instructions for obtaining the certificate that will be used by the FCS_COP.1(2) mechanism to ensure that a signed update has been received from the certificate owner. This may be supplied with the product initially, or may be obtained by some other means.

2. Instructions for obtaining the update itself. This should include instructions for making the update accessible to the TOE (e.g., placement in a specific directory).

3. Instructions for initiating the update process, as well as discerning whether the process was successful or unsuccessful. This includes generation of the hash/digital signature.

The TOE will likely contain security functionality that does not fall in the scope of evaluation under this PP. The operational guidance shall make it clear to an administrator which security functionality is covered by the evaluation activities.

5.2.2.2 Preparative procedures (AGD_PRE.1)

AGD_PRE.1.1d

The developer shall provide the TOE including its preparative procedures.

AGD_PRE.1.1c

The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.

AGD_PRE.1.2c

The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.

AGD_PRE.1.1e

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AGD_PRE.1.2e

The evaluator shall apply the preparative procedures to confirm that the TOE can be prepared securely for operation.

Component Assurance Activity:

As indicated in the introduction above, there are significant expectations with respect to the documentation—especially when configuring the operational environment to support TOE functional requirements. The evaluator shall check to ensure that the guidance provided for the TOE adequately addresses all platforms claimed for the TOE in the ST.

5.2.3 Life-cycle support (ALC)

5.2.3.1 Labelling of the TOE (ALC_CMC.1)

ALC_CMC.1.1d
The developer shall provide the TOE and a reference for the TOE.

ALC_CMC.1.1c
The TOE shall be labelled with its unique reference.

ALC_CMC.1.1e
The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

Component Assurance Activity:
The evaluator shall check the ST to ensure that it contains an identifier (such as a product name/version number) that specifically identifies the version that meets the requirements of the ST. Further, the evaluator shall check the AGD guidance and TOE samples received for testing to ensure that the version number is consistent with that in the ST. If the vendor maintains a web site advertising the TOE, the evaluator shall examine the information on the web site to ensure that the information in the ST is sufficient to distinguish the product.

5.2.3.2 TOE CM coverage (ALC_CMS.1)

ALC_CMS.1.1d
The developer shall provide a configuration list for the TOE.

ALC_CMS.1.1c
The configuration list shall include the following: the TOE itself; and the evaluation evidence required by the SARs.

ALC_CMS.1.2c
The configuration list shall uniquely identify the configuration items.

ALC_CMS.1.1e
The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

Component Assurance Activity:
The 'evaluation evidence required by the SARs' in the NDPP is limited to the information in the ST coupled with the guidance provided to administrators and users under the AGD requirements. By ensuring that the TOE is specifically identified and that this identification is consistent in the ST and in the AGD guidance (as done in the assurance activity for ALC_CMC.1), the evaluator implicitly confirms the information required by this component.

5.2.4 Tests (ATE)

5.2.4.1 Independent testing - conformance (ATE_IND.1)

ATE_IND.1.1d
The developer shall provide the TOE for testing.

ATE_IND.1.1c
The TOE shall be suitable for testing.

ATE_IND.1.1e
The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.1.2e
The evaluator shall test a subset of the TSF to confirm that the TSF operates as specified.

Component Assurance Activity:

The evaluator shall prepare a test plan and report documenting the testing aspects of the system. The test plan covers all of the testing actions contained in the CEM and the body of the NDPP's Assurance Activities. While it is not necessary to have one test case per test listed in an Assurance Activity, the evaluator must document in the test plan that each applicable testing requirement in the ST is covered.

The test plan identifies the platforms to be tested, and for those platforms not included in the test plan but included in the ST, the test plan provides a justification for not testing the platforms. This justification must address the differences between the tested platforms and the untested platforms, and make an argument that the differences do not affect the testing to be performed. It is not sufficient to merely assert that the differences have no affect; rationale must be provided. If all platforms claimed in the ST are tested, then no rationale is necessary.

The test plan describes the composition of each platform to be tested, and any setup that is necessary beyond what is contained in the AGD documentation. It should be noted that the evaluator is expected to follow the AGD documentation for installation and setup of each platform either as part of a test or as a standard pre-test condition. This may include special test drivers or tools. For each driver or tool, an argument (not just an assertion) should be provided that the driver or tool will not adversely affect the performance of the functionality by the TOE and its platform. This also includes the configuration of the cryptographic engine to be used. The cryptographic algorithms implemented by this engine are those specified by the NDPP and used by the cryptographic protocols being evaluated (IPsec, TLS/HTTPS, SSH).

The test plan identifies high-level test objectives as well as the test procedures to be followed to achieve those objectives. These procedures include expected results. The test report (which could just be an annotated version of the test plan) details the activities that took place when the test procedures were executed, and includes the actual results of the tests. This shall be a cumulative account, so if there was a test run that resulted in a failure; a fix installed; and then a successful re-run of the test, the report would show a 'fail' and 'pass' result (and the supporting details), and not just the 'pass' result.

5.2.5 Vulnerability assessment (AVA)

5.2.5.1 Vulnerability survey (AVA_VAN.1)

AVA_VAN.1.1d

The developer shall provide the TOE for testing.

AVA_VAN.1.1c

The TOE shall be suitable for testing.

AVA_VAN.1.1e

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_VAN.1.2e

The evaluator shall perform a search of public domain sources to identify potential vulnerabilities in the TOE.

AVA_VAN.1.3e

The evaluator shall conduct penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Basic attack potential.

Component Assurance Activity:

As with ATE_IND, the evaluator shall generate a report to document their findings with respect to this requirement. This report could physically be part of the overall test report mentioned in

ATE_IND, or a separate document. The evaluator performs a search of public information to determine the vulnerabilities that have been found in network infrastructure devices and the implemented communication protocols in general, as well as those that pertain to the particular TOE. The evaluator documents the sources consulted and the vulnerabilities found in the report. For each vulnerability found, the evaluator either provides a rationale with respect to its non-applicability, or the evaluator formulates a test (using the guidelines provided in ATE_IND) to confirm the vulnerability, if suitable. Suitability is determined by assessing the attack vector needed to take advantage of the vulnerability. For example, if the vulnerability can be detected by pressing a key combination on boot-up, a test would be suitable at the assurance level of the NDPP. If exploiting the vulnerability requires expert skills and an electron microscope, for instance, then a test would not be suitable and an appropriate justification would be formulated.

5.3 Requirement Dependency Rationale

As can be seen in the following table all of the SFR and SAR dependencies are satisfied in this ST.

ST Requirement	CC Dependencies	ST Dependencies
FAU_GEN.1	FPT_STM.1	FPT_STM.1
FAU_GEN.2	FAU_GEN.1 and FIA_UID.1	FAU_GEN.1 and FIA_UIA_EXT.1
FAU_STG_EXT.1	FAU_GEN.1	FAU_GEN.1
FCS_CKM.1	(FCS_CKM.2 or FCS_COP.1) and FCS_CKM.4	FCS_COP.1(*) and FCS_CKM_EXT.4
FCS_CKM_EXT.4	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1)	FCS_CKM.1
FCS_COP.1(1)	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1) and FCS_CKM.4	FCS_CKM.1 and FCS_CKM_EXT.4
FCS_COP.1(2)	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1) and FCS_CKM.4	FCS_CKM.1 and FCS_CKM_EXT.4
FCS_COP.1(3)	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1) and FCS_CKM.4	FCS_CKM.1 and FCS_CKM_EXT.4
FCS_COP.1(4)	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1) and FCS_CKM.4	FCS_CKM.1 and FCS_CKM_EXT.4
FCS_HTTPS_EXT.1	FCS_TLS_EXT.1	FCS_TLS_EXT.1
FCS_RBG_EXT.1	none	none
FCS_SSH_EXT.1	FCS_COP.1	FCS_COP.1(*)
FCS_TLS_EXT.1	FCS_COP.1	FCS_COP.1(*)
FDP_RIP.2	none	none
FIA_PMG_EXT.1	none	none
FIA_UAU.7	FIA_UAU.1	FIA_UIA_EXT.1
FIA_UAU_EXT.2	none	none
FIA_UIA_EXT.1	none	none
FMT_MTD.1	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.2 and FMT_SMF.1
FMT_SMF.1	none	none
FMT_SMR.2	FIA_UID.1	FIA_UIA_EXT.1
FPT_APW_EXT.1	none	none
FPT_SKP_EXT.1	none	none
FPT_STM.1	none	none
FPT_TST_EXT.1	none	none
FPT_TUD_EXT.1	none	none
FTA_SSL.3	none	none
FTA_SSL.4	none	none
FTA_SSL_EXT.1	none	none
FTA_TAB.1	none	none
FTP_ITC.1	none	none

FTP_TRP.1	none	none
ADV_FSP.1	none	none
AGD_OPE.1	ADV_FSP.1	ADV_FSP.1
AGD_PRE.1	none	none
ALC_CMC.1	ALC_CMS.1	ALC_CMS.1
ALC_CMS.1	none	none
ATE_IND.1	ADV_FSP.1 and AGD_OPE.1 and AGD_PRE.1	ADV_FSP.1 and AGD_OPE.1 and AGD_PRE.1
AVA_VAN.1	ADV_FSP.1 and AGD_OPE.1 and AGD_PRE.1	ADV_FSP.1 and AGD_OPE.1 and AGD_PRE.1

Table 4 Requirement Dependencies

6. TOE Summary Specification

This chapter describes the security functions:

- Security audit
- Cryptographic support
- User data protection
- Identification and authentication
- Security management
- Protection of the TSF
- TOE access
- Trusted path/channels

6.1 Security Audit

The TOE is designed to produce syslog conformant messages in a number of circumstances including warnings about the device itself (such as temperature, power failures, etc.) as well as security relevant events (the success and failure login of the user, regardless of the authentication mechanism; changing a user's password; and adding and deleting user accounts). In each case the audit record includes the time and date, identification of the responsible subject (e.g., by network address or user ID), the type of event, the outcome of the event, and other information depending on the event type.

The audit records are stored in a log (internal to the TOE appliance) that is protected so that only an authorized TOE User can read (for which tools accessible via the CLI and Web Management Interface are provided). The protection results from the fact that the logs can be accessed only after a user logs in (see section 6.4 below).

The log stores up to 50 entries after which the audit entries will be overwritten, oldest first. The administrator (with Super User privilege) can (and should) choose to configure one or more external syslog servers where the TOE will simultaneously send a copy of the audit records. The TOE can be configured to use TLS (using any of the four supported, mandatory ciphersuites) to protect audit logs exported to an external server.

The TOE includes a hardware clock that is used to provide reliable time information for the audit records it generates.

The Security audit function is designed to satisfy the following security functional requirements:

- FAU_GEN.1: The TOE can generate audit records for events include starting and stopping the audit function, administrator commands, and all other events identified in **Table 2**. Furthermore, each audit record identifies the date/time, event type, outcome of the event, responsible subject/user, as well as the additional event-specific content indicated in **Table 2**.
- FAU_GEN.2: The TOE identifies the responsible user for each event based on the specific administrator or network entity (identified by IP address) that caused the event.
- FAU_STG_EXT.1: The TOE can be configured to export audit records to an external SYSLOG server. This communication is protected with the use of TLS.

6.2 Cryptographic support

The TOE includes a FIPS 140 certified crypto module providing supporting cryptographic functions. The evaluated configuration requires that the TOE be configured in Common Criteria mode to ensure FIPS certified functions are used.

The following functions have been FIPS certified in accordance with the identified standards.

Functions	Standards	Cert MLXe MR2	 CER 2000 CES 2000
Encryption/Decryption			
• AES CBC (128 and 256 bits)	FIPS Pub 197 NIST SP 800-38A	2717	2715
Cryptographic signature services			
• RSA Digital Signature Algorithm (rDSA) (modulus 2048)	FIPS Pub 186-2	1413	1411
Cryptographic hashing			
• SHA-1, SHA-256, SHA-384, and SHA-512 (digest sizes 160, 256, 384, and 512 bits)	FIPS Pub 180-3	2282	2280
Keyed-hash message authentication			
• HMAC-SHA-1(digest size 160)	FIPS Pub 198-1 FIPS Pub 180-3	1696	1694
Random bit generation			
• CTR_DRBG with sw based noise sources with a minimum of 256 bits of non-determinism	NIST SP 800-90	454	452
Key Derivation Functions			
• TLS and SSH	NIST SP 800-135	175	173

Table 5 Cryptographic Functions

While the TOE generally fulfills all of the NIST SP 800-56B requirements without extensions, the following table specifically identifies the "should", "should not", and "shall not" conditions from the publication along with an indication of how the TOE conforms to those conditions.

NIST SP800-56B Section Reference	"should", "should not", or "shall not"	Implemented?	Rationale for deviation
5.6	Should	Yes	Not applicable
5.8	shall not	No	Not applicable
5.9	shall not (first occurrence)	No	Not applicable
5.9	shall not (second occurrence)	No	Not applicable
6.1	should not	No	Not applicable
6.1	should (first occurrence)	Yes	Not applicable
6.1	should (second occurrence)	Yes	Not applicable
6.1	should (third occurrence)	Yes	Not applicable
6.1	should (fourth occurrence)	Yes	Not applicable
6.1	shall not (first occurrence)	No	Not applicable
6.1	shall not (second occurrence)	No	Not applicable
6.2.3	Should	Yes	Not applicable
6.5.1	Should	Yes	Not applicable

NIST SP800-56B Section Reference	"should", "should not", or "shall not"	Implemented?	Rationale for deviation
6.5.2	Should	Yes	Not applicable
6.5.2.1	Should	Yes	Not applicable
6.6	shall not	No	Not applicable
7.1.2	Should	Yes	Not applicable
7.2.1.3	Should	Yes	Not applicable
7.2.1.3	should not	No	Not applicable
7.2.2.3	should (first occurrence)	Yes	Not applicable
7.2.2.3	should (second occurrence)	Yes	Not applicable
7.2.2.3	should (third occurrence)	Yes	Not applicable
7.2.2.3	should (fourth occurrence)	Yes	Not applicable
7.2.2.3	should not	No	Not applicable
7.2.2.3	shall not	No	Not applicable
7.2.3.3	should (first occurrence)	Yes	Not applicable
7.2.3.3	should (second occurrence)	Yes	Not applicable
7.2.3.3	should (third occurrence)	Yes	Not applicable
7.2.3.3	should (fourth occurrence)	Yes	Not applicable
7.2.3.3	should (fifth occurrence)	Yes	Not applicable
7.2.3.3	should not	No	Not applicable
8	Should	Yes	Not applicable
8.3.2	should not	No	Not applicable

Table 6 NIST SP800-56B Conformance

The TOE provides RFC compliant TLS, HTTPS, and SSH implementations without any optional components.

The TOE uses a software-based random bit generator that complies with Special Publication 800-90 using CTR_DRBG when operating in the FIPS mode. AES-256 is used in conjunction with a minimum of 256 bits of entropy accumulated from the processing stack, hardware serial numbers, and the low-order bits from the current time of day.

The TOE supports the following secret keys, private keys and CSPs:

Key or CSP:	Zeroized upon:	Stored in:	Zeroized by:
SSH host RSA private key	Command	Flash	Overwriting once with zeros
SSH host RSA public key	Command	Flash	Overwriting once with zeros
SSH client RSA public key	Command	Flash	Overwriting once with zeros
SSH session key	End of session	RAM	Overwriting once with zeros
TLS host RSA private key	Command	Flash	Overwriting once with zeros
TLS host RSA digital certificate	Command	Flash	Overwriting once with zeros
TLS pre-master secret	Handshake done	RAM	Overwriting once with zeros
TLS session key	Close of session	RAM	Overwriting once with zeros
DH Private Exponent	New key exchange	RAM	Overwritten with new value
DH Public Key	Not applicable	RAM	Public value
User Password	Command	Flash	Overwriting once with zeros
Port Administrator Password	Command	Flash	Overwriting once with zeros
Crypto Officer Password	Command	Flash	Overwriting once with zeros
TACACS+ Secret	Command	Flash	Overwriting once with zeros
Firmware Integrity / Load RSA public key	Not applicable	Flash	Public value
DRBG Seed	Every 100ms	RAM	Overwritten with new value
DRBG Value V	Every 100ms	RAM	Overwritten with new value

Key or CSP:	Zeroized upon:	Stored in:	Zeroized by:
DRBG Constant C	Every 100ms	RAM	Overwritten with new value

Table 7 Keys and CSPs

The TOE stores all persistent secret and private keys in FLASH and store all ephemeral keys in RAM (as indicated in the above table). Additionally, the TOE is designed to zeroize secret and private keys when they are no longer required by the TOE as detailed below. The TOE's zeroization has been subjected to FIPS 140 validation. Note that zeroization occurs as follows: 1) when deleted from FLASH, the previous value is overwritten once with zeroes; 2) when added or changed in FLASH, any old value is overwritten completely with the new value; and, 3) the zeroization of values in RAM is achieved by overwriting once with zeroes.

The TOE supports the following different zeroization methods for its secret keys, private keys and CSPs (note that no public keys appear in this list; they are public and thus need not be zeroizeable). For any given CSP in the table above, there may be multiple zeroization methods available.

- command: *fips zeroize all* - The device zeros out the shared secrets use by various networking protocols including host access passwords, SSH host keys, and HTTPS host keys with the digital signature.
- command: *no fips enable* or *no fips enable common-criteria* - Zeroizes shared secrets, SSH and HTTPS host keys, and the HTTPS certificate based on the configured FIPS policy. Either of these commands will take the TOE out of its evaluated configuration and zeroize the secrets assuming a default FIPS policy. An administrator can use the prior command, fips zeroize all, to conclusively zeroize all CSPs, secret, and private keys, irrespective of the configured FIPS policy.
- The SSH session key is transient. It zeroized at the end of a session and recreated at the beginning of a new session.
- The TLS pre-master secret is generated during the TLS handshake. It is destroyed after it is used.
- The TLS session key is generated for every HTTPS session. The TLS session key is deleted after the session is closed.
- The DRBG seed is recomputed periodically on 100 millisecond intervals.
- The DH private exponent is generated at the beginning of DH KEX. A new random number overwrites the memory location used to store the value each time a new session is initiated.
- For SSH, the RSA private key is stored in a locally generated file on flash during the key generation process. The file is removed during zeroization. The *crypto key zeroize* command removes the keys.
- For TLS, the RSA private key is stored in a locally generated file on flash during the key generation process. The private and public key data is overwritten with space characters during zeroization. The *crypto-ssl zeroize* command zeroes out the RSA key pair.

These supporting cryptographic functions are included to support the SSHv2 (compliant with RFCs 4251, 4252, 4253, and 4254) and TLSv1.0 (compliant with RFC 2246)/HTTPS (compliant with RFC 2818) secure communication protocols.

The TOE supports TLSv1.0, TLS 1.1,and TLS 1.2 with AES (CBC) 128 or 256 bit ciphers, in conjunction with SHA-1, and RSA. The following cipher suites are implemented by the TOE: TLS_RSA_WITH_AES_128_CBC_SHA, TLS_RSA_WITH_AES_256_CBC_SHA, TLS_DHE_RSA_WITH_AES_128_CBC_SHA, TLS_DHE_RSA_WITH_AES_256_CBC_SHA, ~~TLS_RSA_WITH_AES_128_CBC_SHA256, TLS_RSA_WITH_AES_256_CBC_SHA256, TLS_DHE_RSA_WITH_AES_128_CBC_SHA256, TLS_DHE_RSA_WITH_AES_256_CBC_SHA256, TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256, TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384, TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256, and TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384.~~

The TOE supports SSHv2 with AES (CBC) 128 or 256 bit ciphers, in conjunction with HMAC-SHA-1, and RSA (with diffie-hellman-group14-sha1 for the key exchange method). While other ciphers and hashes are implemented in the product, they are disabled while the TOE is operating in Common Criteria or FIPS mode.

The TOE allows users to perform SSHv2 authentication using password based authentication and allows users to upload a public key for SSHv2 public key client authentication. The TOE's SSHv2 implementation limits SSH packets to a size of 256K bytes. Whenever the timeout period or authentication retry limit is reached, the TOE closes the applicable TCP connection and releases the SSH session resources. As SSH packets are being received, the TOE uses a buffer to build all packet information. Once complete, the packet is checked to ensure it can be appropriately decrypted. However, if it is not complete when the buffer becomes full (256K bytes) the packet will be dropped and the connection terminated.

The Cryptographic support function is designed to satisfy the following security functional requirements:

- FCS_CKM.1: See Table 6 NIST SP800-56B Conformance above.
- FCS_CKM_EXT.4: Keys are zeroized when they are no longer needed by the TOE.
- FCS_COP.1(1): See Table 5 Cryptographic Functions above.
- FCS_COP.1(2): See Table 5 Cryptographic Functions above.
- FCS_COP.1(3): See Table 5 Cryptographic Functions above.
- FCS_COP.1(4): See Table 5 Cryptographic Functions above.
- FCS_HTTPS_EXT.1: The TOE supports TLS/HTTPS web-based secure administrator sessions.
- FCS_RBG_EXT.1: See Table 5 Cryptographic Functions above.
- FCS_SSH_EXT.1: The TOE supports SSHv2 interactive command-line secure administrator sessions as indicated above.
- FCS_TLS_EXT.1: The TOE supports TLS/HTTPS web-based secure administrator sessions.

6.3 User data protection

The TOE is designed to ensure its own internal integrity as well as to protect user data from potential, unintended reuse by clearing resources (e.g., memory) as they are allocated to create objects used in the implementation of the TOE operations. Note that volatile memory is the primary resource involved in normal TOE execution while its persistent storage is based on non-volatile flash memory.

When the TOE sends a network packet, it must request a buffer from the buffer pool. After using a buffer, the TOE releases the buffer back to the buffer pool. In response to a request, the buffer pool will return a buffer and its length, where the length is greater than or equal to that requested. The TOE will compare the length of the returned buffer to that which it requested (the size of the packet), overwrite the returned buffer with packet data (destroying any residual data present in the buffer), and, if the provided buffer exceeds the requested size of the packet, overwrite any extra space with zeros (thus ensuring that no residual data can leak from the TOE).

The User data protection function is designed to satisfy the following security functional requirements:

- FDP_RIP.2: The TOE always overwrites resources when allocated for use in objects.

6.4 Identification and authentication

The TOE requires users to be identified and authenticated before they can use functions mediated by the TOE, except to display a message of the day banner and to permit network traffic to flow through the TOE without identification or authentication. The network routing services that the TOE allows includes network traffic being routed through the TOE as well as network routing protocol traffic destined to the TOE (including DNS, ARP, ICMP, BootP, DHCP, RIP, OSPF, BGP, VRRP, VRRP-E, Multi-VRF) but does not include any management configuration of the TOE's network routing services. The TOE authenticates TOE Users against their user name, password and privilege level.

The Authorized Administrator with Super User privilege represents the "administrator" referred to in the security requirements of the protection profile. Other accounts with privileges other than Super User were not testing during evaluation. The Authorized Administrator with Super User privilege defines local user (or TOE User) accounts and to assign passwords and privilege levels to the accounts. Each user account has a user name, password, and a privilege level associated with it. There is a default privilege level account associated with each privilege level and each has its own password. It is up to the Authorized Administrator with Super User privilege to decide whether or how to use these legacy accounts. Note however, that each has an identity, password, and privilege level.

The user roles offered by the TOE are categorized differently when described in FIPS documentation. Specifically, the Authorized Administrator with Super User privilege equates to the FIPS Crypto Officer Role, the Port Configuration User equates to the FIPS Port Configuration Administrator Role (and has write access to the interface configuration mode only), and a user with read-only privileges and no configuration mode access equates to the FIPS User Role.

While the Authorized Administrator with Super User privilege can create or otherwise modify accounts freely, other users cannot change their own (or any other) security attributes. Note that the TOE supports a password enforcement configuration where the minimum password length can be set by an administrator up to 48 characters. Passwords can be created using any alphabetic, numeric, and a wide range of special characters (identified in FIA_PMG_EXT.1).

Additional authentication mechanisms can also be configured by an Authorized Administrator using an Authentication Method List. This allows some flexibility in setting up authentication mechanisms when desired. The available mechanisms include the Local Password for the Super User Privilege level, TACACS+ authentication, and the SSH public key authentication mechanism. An administrator can create users, associate passwords with user accounts, and can also set the privilege level associated with a user. User's after authenticating, may upload a public key to be used with SSH client public key authentication. However, the TOE's TOE's TACACS+ implementation does not support SSH client public key authentication (the TOE supports SSH client public key authentication through public keys stored locally within the TOE). Additionally, the TOE's Web Management Interface (present in the MLX Series) does not support TACACS+ authentication of users. When authentication succeeds, the TOE looks up the user's defined privilege level, assigns that to the user's session, and presents the user with a command prompt (the "#" character, e.g., "`Brocade(config)#`").

The Identification and authentication function is designed to satisfy the following security functional requirements:

- FIA_PMG_EXT.1: The TOE implements a rich set of password composition constraints as described above.
- FIA_UAU.7: The TOE does not echo passwords as they are entered; rather '*' characters are echoed when entering passwords.
- FIA_UAU_EXT.2: The TOE can be configured to utilize local password-based authentication and SSH public-key-based authentication mechanisms.
- FIA_UIA_EXT.1: The TOE doesn't offer any services or access to its functions, except for the switching/routing of network traffic and displaying a message of the day banner, without requiring a user to be identified and authenticated.

6.5 Security management

The TOE associates each defined user account with a privilege level. The most privileged level is Super User (with regards to the requirements in this Security Target users with lesser privilege levels are referred to collectively simply as TOE users since such users do not have complete read-and-write access to the system). Again, as stated in section 6.4, other accounts with privileges other than Super User were not tested during the evaluation. The TOE implements an internal access control mechanism that bases decisions about the use of functions and access to TOE data on those privilege levels. In this manner, the TOE is able to ensure that only the Authorized Administrator with Super User privilege can access audit configuration data, information flow policy ACLs, user and administrator security attributes (including passwords and privilege levels), authentication method lists, the logon failure threshold, the remote access user list; and cryptographic support settings.

Other than the Super User level, the TOE implements a Read Only level where only basic commands can be issued and no changes can be made and a Port Configuration level where non-security device parameters can be managed. Collectively, this ST refers to all users of the TOE as "TOE Users" where the "Authorized Administrator with Super User privilege" is a subset of that broader role.

The TOE offers command line functions which are accessible via the CLI. The CLI is a text based interface which can be accessed from a directly connected terminal or via a remote terminal using SSH. These command line functions can be used to effectively manage every security policy, as well as the non-security relevant aspects of the TOE.

Similarly, the TOE's MLX series offers a Web Management Interface that offers access to the same functions as the CLI. While the Web Management Interface could be configured to be accessible via HTTP or HTTPS (using TLSv1.0, 1.1, and 1.2), the evaluated configuration only includes the use of HTTPS (note that the TOE does not support client authentication) to ensure that the administrative session is not subject to modification or disclosure.

The following table provides the list of security-related commands used to configure or examine the TOE security settings. The services listed here reflect the minimal set needed to properly configure the TOE to comply with the requires of the *Protection Profile for Network Devices*, version 1.1, 8 June 2012 (NDPP) with Errata #3, 3 November 2014.

Command	Tested Command Variantts	Description
write	write memory	Write to persistent storage
crypto	crypto key generate	Invoke cryptographic functions
openssl	openssl s_server	Configure secure connections (e.g., with syslog)
logging	logging host <ip-address> ssl-port <port>	Configure the audit logging host
reload	reload	Reload the current flash image
console	console timeout <time>	Manage console properties
banner	banner motd +	Manage the login banner
exit	exit	Logout or exit current session
ntp	ntp	Switch to ntp configuration mode
config	config t	Switch to configuration mode
username	username <user> password	Manage user accounts
clock	clock set <time>	Manage the internal clock
server	server <ntp server ip> minpoll <time>	Configure external services
crypto-ssl	crypto-ssl certificate generate	Manage web server properties
web-management	web-management session-timeout <time>	Manage web interface
fips	fips enable common-criteria fips show fips zeroize all	Manage FIPS and Common Criteria configuration
ip	ip ssh pub-key-file ip ssh idle-time <time>	Manage ip connection (e.g., ssh) configuration
aaa	aaa authentication aaa authentication enable default tacacs+ local aaa authentication login default tacacs+ local aaa authentication web-server default local	Configure the aaa authentication functions
tacacs-server	tacacs-server host <ipaddr> ssl-auth-port <port> default tacacs-server retransmit <retransmit period> tacacs-server timeout <timeout period> tacacs-server key <key>	Configure TACACAS+ server
enable	enable aaa enable password-min-length 15 enable user password-masking	Enable console login features
show	show flash show ver	Show identified configuration information

Command	Tested Command Variantts	Description
	show clock show ip client-pub-key show ip ssl show logging show run | <options>	

Table 8 Security Related Configuration Commands

The TOE also provides a comprehensive set of network routing configuration commands. These commands were not exercised as the above services in Table 8 represent the minimum set of commands needed to for proper configuration.

The Security management function is designed to satisfy the following security functional requirements:

- FMT_MTD.1: The TOE restricts the access to manage TSF data that can affect the security functions of the TOE to Authorized Administrator with Super User privilege (aka Security Administrator).
- FMT_SMF.1: The TOE includes the functions necessary to enable/disable available network services, to manage the cryptomodule and associated functions, and to manage and verify updates of the TOE software and firmware.
- FMT_SMR.2: The TOE includes roles associated with privileges. 'Authorized Administrator with Super User privilege' corresponds to the required 'Authorized Administrator' also referred to as 'Security Administrator' in some requirements.

6.6 Protection of the TSF

The TOE is an appliance and as such is designed to work independent of other components to a large extent. Secure communication with third-party peers as addressed in section 6.8, Trusted path/channels, and secure communication among multiple instances of the TOE is limited to a direct link between clustered switch appliances. Normally clustered components are co-located and connected via a link that would not be exposed outside of the same physical environment. As such, no additional protection (e.g., encryption) should be necessary in most operational environments.

While the administrative interface is function rich, the TOE is designed specifically to not provide access to locally stored passwords (which are protected using MD-5 hashing) and also, while cryptographic keys can be entered, the TOE does not disclose any cryptographic keys stored in the TOE. The TOE is a hardware appliance that includes a hardware-based real-time clock. The TOE's embedded OS manages the clock and exposes administrator clock-related functions. The TOE can be configured to periodically synchronize its clock with a time server, but the TOE can only ensure its own reliability and not that of an external time mechanism. The TOE also implements the timing elements through timeout functionality due to inactivity for terminating both local and remote sessions. Note that the clock is used primarily to provide timestamp for audit records, but is also used to supporting timing elements of cryptographic functions.

The TOE includes a number of built in diagnostic tests that are run during start-up to determine whether the TOE is operating properly. An administrator can configure the TOE to reboot or to stop, with errors displayed, when an error is encountered. When operating in FIPS mode, the power-on self-tests comply with the FIPS 140-2 requirements for self-testing. The module performs Cryptographic algorithm known answer tests, firmware integrity tests using RSA signature verification and conditional self-tests for DRBG, Hardware RNG, Pair-wise consistency tests on generation of RSA keys, and a Firmware load test (RSA signature verification). Upon failing any of its FIPS mode power-on self-tests, the TOE will refuse to boot.

The TOE supports loading a new software image manually by the administrator using CLI commands. From the CLI, an administrator can use SCP in order to download a software image, and the TOE, prior to actually installing and using the new software image, will verify its digital certificate using the public key in the certificate configured in the TOE. An unverified image cannot be installed. Note that the TOE comes preinstalled with an applicable Brocade public certificate.

The Protection of the TSF function is designed to satisfy the following security functional requirements:

- FPT_SKP_EXT.1: The TOE does not offer any functions that will disclose to any users a stored cryptographic key.
- FPT_APW_EXT.1: The TOE does not offer any functions that will disclose to any user a plain text password. Furthermore, locally defined passwords are not stored in plaintext form.
- FPT_STM.1: The TOE includes its own hardware clock.
- FPT_TST_EXT.1: The TOE includes a number of power-on diagnostics that will serve to ensure the TOE is functioning properly. The tests include ensure memory and flash can be accessed as expected, to ensure that software checksums are correct, and also to test the presence and function of plugged devices.
- FPT_TUD_EXT.1: The TOE provides function to query the version and upgrade the software embedded in the TOE appliance. When installing updated software, digital signatures are used to authenticate the update to ensure it is the update intended and originated by Brocade.

6.7 TOE access

The TOE can be configured to display an administrator-configured message of the day banner that will be displayed before authentication is completed (before the user enters his password). The banner will be displayed when accessing the TOE via the console, SSH, or TLS/HTTPS interfaces.

The TOE can be configured by an administrator to set a session timeout value (any value up to 240 minutes, with 0 disabling the timeout) – the default timeout is disabled. A session (local or remote) that is inactive (i.e., no commands issuing from the remote client) for the defined timeout value will be terminated. Upon exceeding the session timeout (if set), the TOE logs the user off, but leaves the user's console displaying the last contents.

The user will be required to login in after any session has been terminated due to inactivity or after voluntary termination. Of course, administrators can logout of local or remote sessions at any time.

The TOE access function is designed to satisfy the following security functional requirements:

- FTA_SSL.3: The TOE terminates remote sessions that have been inactive for an administrator-configured period of time.
- FTA_SSL.4: The TOE provides the function to logout (or terminate) the both local and remote user sessions as directed by the user.
- FTA_SSL_EXT.1: The TOE terminates local sessions that have been inactive for an administrator-configured period of time.
- FTA_TAB.1: The TOE can be configured to display administrator-defined advisory banners when administrators successfully establish interactive sessions with the TOE, allowing administrators to terminate their session prior to performing any functions.

6.8 Trusted path/channels

The TOE implements SSHv2 and HTTPS (using TLSv1.2) which are required to be used for remote administration. When an administrator attempts to connect to the TOE, the TOE attempts to negotiate a session. If the session cannot be negotiated, the connection is dropped.

When a client attempts to connect using SSH or TLS/HTTPS, the TOE and the client will negotiate the most secure algorithms available at both ends to protect that session. SSH_RSA is the only public key authentication algorithm used by the SSH transport implementation, and DH group 14 is the only Diffie-Hellman group the TOE supports when configured in Common Criteria mode.

In each case, AES-CBC with 128-bit or 256-bit keys is implemented for encryption and decryption and RSA using up to 2048-bit keys are implemented for key exchange and authentication (i.e., distribution).

Note that the product includes other cryptographic algorithms, but since they are not FIPS certified they are not recommended for use and excluded from the scope of evaluation.

Remote connection to SYSLOG servers is protected using TLS (as specified earlier).

In all cases, the endpoints are assured by virtue of the certificates installed, trusted, and reviewable when connecting and by virtue of user authentication.

The TOE update service is secured using SCP, as when operating in FIPS (or Common Criteria) Mode, the TOE prevents the use of TFTP to retrieve a new TOE firmware image.

The Trusted path/channels function is designed to satisfy the following security functional requirements:

- FTP_ITC.1: In the evaluated configuration, the TOE must be configured to use TLS to ensure that any authentication operations and exported audit records are sent only to the configured server so they are not subject to inappropriate disclosure or modification.
- FTP_TRP.1: The TOE provides SSH and TLS/HTTPS, based on its embedded cryptomodule, to ensure secure remote administration. In each case, the administrator can initiate the remote session, the remote session is secured (disclosure and modification) using FIPS certified cryptographic operations, and all remote security management functions require the use of one of these secure channels.