
Brocade Communications Systems, Inc. MLX and NetIron CER 2000 Series Router Security Target

Version 1.0
1 May 2014

Prepared for:
Brocade Communications Systems, Inc.

130 Holger Way
San Jose, CA 95134

Prepared By:



Leidos Inc (formerly Science Applications International Corporation)

Common Criteria Testing Laboratory
6841 Benjamin Franklin Drive, Columbia, Maryland 21046

1. SECURITY TARGET INTRODUCTION	4
1.1 SECURITY TARGET, TOE AND CC IDENTIFICATION	4
1.2 CONFORMANCE CLAIMS	5
1.3 CONVENTIONS AND TERMINOLOGY	5
1.3.1 Conventions	5
1.3.2 Terminology	6
1.3.3 Acronyms	6
2. TOE DESCRIPTION	7
2.1 TOE OVERVIEW	9
2.2 TOE ARCHITECTURE	9
2.2.1 Physical Boundaries	9
2.2.2 Logical Boundaries	10
2.3 TOE DOCUMENTATION	11
3. SECURITY PROBLEM DEFINITION	13
3.1 ORGANIZATIONAL POLICIES	13
3.2 THREATS	13
3.3 ASSUMPTIONS	13
4. SECURITY OBJECTIVES	15
4.1 SECURITY OBJECTIVES FOR THE TOE	15
4.2 SECURITY OBJECTIVES FOR THE ENVIRONMENT	15
5. IT SECURITY REQUIREMENTS	17
5.1 EXTENDED REQUIREMENT DEFINITIONS	17
5.2 TOE SECURITY FUNCTIONAL REQUIREMENTS	17
5.2.1 Security audit (FAU)	18
5.2.2 Cryptographic support (FCS)	21
5.2.3 User data protection (FDP)	28
5.2.4 Identification and authentication (FIA)	28
5.2.5 Security management (FMT)	30
5.2.6 Protection of the TSF (FPT)	31
5.2.7 TOE access (FTA)	33
5.2.8 Trusted path/channels (FTP)	34
5.3 TOE SECURITY ASSURANCE REQUIREMENTS	36
5.3.1 Development (ADV)	36
5.3.2 Guidance documents (AGD)	36
5.3.3 Life-cycle support (ALC)	38
5.3.4 Tests (ATE)	39
5.3.5 Vulnerability assessment (AVA)	39
6. TOE SUMMARY SPECIFICATION	41
6.1 SECURITY AUDIT	41
6.2 CRYPTOGRAPHIC SUPPORT	42
6.3 USER DATA PROTECTION	45
6.4 IDENTIFICATION AND AUTHENTICATION	45
6.5 SECURITY MANAGEMENT	46
6.6 PROTECTION OF THE TSF	46
6.7 TOE ACCESS	47
6.8 TRUSTED PATH/CHANNELS	48
7. PROTECTION PROFILE CLAIMS	48
8. RATIONALE	49
8.1 SECURITY OBJECTIVES RATIONALE	49

8.1.1	<i>Security Objectives Rationale for the TOE and Environment</i>	49
8.2	SECURITY REQUIREMENTS RATIONALE.....	51
8.2.1	<i>Security Functional Requirements Rationale</i>	52
8.3	SECURITY ASSURANCE REQUIREMENTS RATIONALE.....	55
8.4	REQUIREMENT DEPENDENCY RATIONALE.....	55
8.5	TOE SUMMARY SPECIFICATION RATIONALE.....	56

LIST OF TABLES

Table 1	TOE Security Functional Components	18
Table 2	Auditable Events	20
Table 3	EAL 1 Assurance Components	36
Table 4	Cryptographic Functions	42
Table 5	NIST SP800-56B Conformance	43
Table 6	Environment to Objective Correspondence	49
Table 7	Objective to Requirement Correspondence	52
Table 8	Requirement Dependencies	56
Table 9	Security Functions vs. Requirements Mapping	57

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 MLX and NetIron CER 2000 Series routers provided by Brocade Communications Systems, Inc.. Each MLX and NetIron CER 2000 router includes a hardware appliance running a version of Brocade's proprietary IronWare Operating System (IOS). Each MLX and NetIron CER 2000 router appliance is designed to manage the flow of network information.

In the context of this evaluation, the TOE is a network device that provides a secure base, primarily involving auditing, cryptographic support (for network communication and update integrity), user identification and authentication, and secure management and product updates, for its other operational functions primarily related to switching and routing IP network traffic.

The Security Target contains the following additional sections:

- TOE Description (Section 2)
- Security (Section 3)
- Security Objectives (Section 4)
- IT Security Requirements (Section 5)
- TOE Summary Specification (Section 6)
- Protection Profile Claims (Section 7)
- Rationale (Section 8).

1.1 Security Target, TOE and CC Identification

ST Title – Brocade Communications Systems, Inc. MLX and NetIron CER 2000 Series Router Security Target

ST Version – Version 1.0

ST Date – 5/1/14

TOE Identification – Brocade MLX and NetIron CER 2000 Router products with IOS 5.3 including the following series and models:

- Brocade MLX Series Hardware Platforms:
 - BR-MLXE-16-MR-M-AC
 - BR-MLXE-16-MR-M-DC
 - BR-MLXE-16-MR2-M-AC
 - BR-MLXE-16-MR2-M-DC
 - BR-MLXE-8-MR-M-AC
 - BR-MLXE-8-MR-M-DC
 - BR-MLXE-8-MR2-M-AC
 - BR-MLXE-8-MR2-M-DC
 - BR-MLXE-4-MR-M-AC
 - BR-MLXE-4-MR-M-DC
 - BR-MLXE-4-MR2-M-AC
 - BR-MLXE-4-MR2-M-DC
- Each of these devices runs the following evaluated software (IOS 5.3), as displayed by the 'show version' CLI command:
 - Boot: Version 5.3.0T165 Copyright (c) 1996-2009 Brocade Communications Systems, Inc.
Compiled on Nov 16 2011 at 10:05:30 labeled as xmpr05300
(517880 bytes) from boot flash
 - Monitor: Version 5.3.0T165 Copyright (c) 1996-2009 Brocade Communications Systems, Inc.
Compiled on Nov 16 2011 at 10:04:52 labeled as xmb05300
(524496 bytes) from code flash
 - IronWare: Version 5.3.0eT163 Copyright (c) 1996-2009 Brocade Communications Systems, Inc.

Compiled on Apr 22 2014 at 22:02:40 labeled as xmr05300ea
(8116989 bytes) from Primary

- 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
- Each of these devices runs the following evaluated software (IOS 5.3), as displayed by the 'show version' CLI command:
 - Boot: Version 5.3.0T185 Copyright (c) 1996-2009 Brocade Communications Systems, Inc.
Compiled on Nov 16 2011 at 10:06:46 labeled as ceb05300
(447585 bytes) from boot flash
 - Monitor: Version 5.3.0T185 Copyright (c) 1996-2009 Brocade Communications Systems, Inc.
Compiled on Nov 16 2011 at 10:06:46 labeled as ceb05300
(447585 bytes) from code flash
 - IronWare: Version 5.3.0eT183 Copyright (c) 1996-2009 Brocade Communications Systems, Inc.
Compiled on Apr 22 2014 at 22:30:18 labeled as ce05300ea
(14496944 bytes) from Primary.

TOE Developer – Brocade Communications Systems, Inc.

Evaluation Sponsor – Brocade Communications Systems, Inc.

CC Identification – Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 3, July 2009

1.2 Conformance Claims

This TOE is conformant to the following CC specifications:

- The ST conforms to the *Security Requirements for Network Devices*, version 1.1, 8 June 2012 (NDPP)
- Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, Version 3.1, Revision 3, July 2009.
 - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1 Revision 3, July 2009.
 - Part 3 Conformant

1.3 Conventions and Terminology

This section specifies the formatting information used in the Security Target.

1.3.1 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 number in parentheses placed at the end of the component. For example FDP_ACC.1(1) and FDP_ACC.1(2) indicate that the ST includes two iterations of the FDP_ACC.1 requirement, (1) and (2).
 - 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.

1.3.2 Terminology

<i>User</i>	Any entity (human or otherwise) outside the TOE that interacts with the TOE.
<i>Unauthorized User</i>	An entity that interacts (or attempts to interact) with the TOE Security Function (TSF) in an unapproved manner.
<i>Authorized Administrator</i>	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.
<i>TOE User</i>	Any person who interacts with the TOE.
<i>External IT entity</i>	Any IT product or system, untrusted or trusted, outside of the TOE that interacts with the TOE.
<i>Role</i>	A predefined set of rules establishing the allowed interactions between a user and the TOE.
<i>Identity</i>	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.
<i>Authentication data</i>	Information used to verify the claimed identity of a user.
<i>Object</i>	An entity within the TOE Security Function (TSF) Scope of Control (TSC) that contains or receives information and upon which subjects perform operations.
<i>Subject</i>	An entity within the TSC that causes operations to be performed.
<i>Authorized User</i>	A user who may, in accordance with the TOE Security Policy (TSP), perform an operation.

1.3.3 Acronyms

ACL	Access Control List
AUT	Authentication

CC	Common Criteria for Information Technology Security Evaluation
CEM	Common Evaluation Methodology for Information Technology Security
CM	Configuration Management
CLI	Command Line Interface
EAL	Evaluation Assurance Level
FDP	User Data Protection CC Class
FIA	Identification and Authentication CC Class
FMT	Security Management CC Class
FSP	Functional Specification
HLD	High Level Design
IOS	IronWare™ operating system
ISO 15408	Common Criteria 2.1 ISO Standard
IT	Information Technology
MOF	Management of Functions
MTD	Management of TSF Data
OSP	Organization Security Policy
PP	Protection Profile
SAR	Security Assurance Requirement
SFP	Security Function Policy
SFR	Security Functional Requirement
SM	Security Management
SMR	Security Management Roles
SOF	Strength of Function
SSH	Secure Shell
ST	Security Target
TACACS	Terminal Access Controller Access Control System
TOE	Target of Evaluation
TSC	TSF Scope of Control
TSF	TOE Security Functions
TSP	TOE Security Policy
UAU	User Authentication
UDP	User Data Protection

2. TOE Description

The Target of Evaluation (TOE) is Brocade MLX and NetIron CER 2000 router products with IOS 5.3 including the following series and models:

- Brocade MLX Series Hardware Platforms:
 - BR-MLXE-16-MR-M-AC
 - BR-MLXE-16-MR-M-DC
 - BR-MLXE-16-MR2-M-AC
 - BR-MLXE-16-MR2-M-DC
 - BR-MLXE-8-MR-M-AC
 - BR-MLXE-8-MR-M-DC
 - BR-MLXE-8-MR2-M-AC
 - BR-MLXE-8-MR2-M-DC
 - BR-MLXE-4-MR-M-AC
 - BR-MLXE-4-MR-M-DC
 - BR-MLXE-4-MR2-M-AC
 - BR-MLXE-4-MR2-M-DC
- Each of these devices runs the following evaluated software (IOS 5.3), as displayed by the ‘show version’ CLI command:

- Boot: Version 5.3.0T165 Copyright (c) 1996-2009 Brocade Communications Systems, Inc. Compiled on Nov 16 2011 at 10:05:30 labeled as xmprm05300 (517880 bytes) from boot flash
- Monitor: Version 5.3.0T165 Copyright (c) 1996-2009 Brocade Communications Systems, Inc. Compiled on Nov 16 2011 at 10:04:52 labeled as xmb05300 (524496 bytes) from code flash
- IronWare: Version 5.3.0eT163 Copyright (c) 1996-2009 Brocade Communications Systems, Inc. Compiled on Apr 22 2014 at 22:02:40 labeled as xmr05300ea (8116989 bytes) from Primary
- 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
- Each of these devices runs the following evaluated software (IOS 5.3), as displayed by the ‘show version’ CLI command:
 - Boot: Version 5.3.0T185 Copyright (c) 1996-2009 Brocade Communications Systems, Inc. Compiled on Nov 16 2011 at 10:06:46 labeled as ceb05300 (447585 bytes) from boot flash
 - Monitor: Version 5.3.0T185 Copyright (c) 1996-2009 Brocade Communications Systems, Inc. Compiled on Nov 16 2011 at 10:06:46 labeled as ceb05300 (447585 bytes) from code flash
 - IronWare: Version 5.3.0eT183 Copyright (c) 1996-2009 Brocade Communications Systems, Inc. Compiled on Apr 22 2014 at 22:30:18 labeled as ce05300ea (14496944 bytes) from Primary.

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/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/downloads/documents/data_sheets/product_data_sheets/NetIron_CER_2000_DS.pdf

While there are several models in each series, they differ primarily in physical form factor, number and types of connections and slots, and relative performance. Each device provides the same security characteristics as claimed in this security target.

2.1 TOE Overview

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 Multiservice IronWare Operating System (IOS). The IOS controls the routing network frames and packets among the connection 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 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. 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
- Flash memory, used to store the operating system image
- Non-volatile memory, which stores configuration parameters used to initialize the system at system startup
- Multiple physical network interfaces either fixed in configuration or removable as in a chassis based product.

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, or dropped in accordance with a configured policy.

2.2 TOE Architecture

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

2.2.1 Physical Boundaries

Each TOE appliance runs a version of the Brocades IOS and has physical network connections to its environment to facilitate routing 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 a syslog server in the environment. This is generally advisable given the limited audit log storage space on the evaluated appliances.

The use of RADIUS and TACACS/TACACS+ external authentication services are excluded from the evaluated configuration of the TOE.

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

2.2.2 Logical Boundaries

The TOE logical boundary consists of the security functionality of Brocade MLX and NetIron CER 2000 router appliances 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 and not modify any security related configuration settings.
2. Web Management Access is excluded from the evaluated configuration.
3. Telnet access is assumed to be **disabled** for remote/network access to the TOE.
4. The *Strict Password Enforcement* setting is assumed to be **enabled** in the evaluated configuration.
5. The TOE must be operated in FIPS Common Criteria mode (i.e. fips enable common-criteria command)..

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.

2.2.2.1 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 on the network.

2.2.2.2 Cryptographic support

The TOE includes a 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.

2.2.2.3 User data protection

The TOE performs a wide variety of network 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.

2.2.2.4 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.

The TOE also provides the authorized administrators with the ability to configure Authentication Method lists. These lists are used to specify the order in which the authentication methods are employed whenever there are one or more authentication methods available.

2.2.2.5 Security management

The TOE provides Command Line Interface (CLI) commands 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.

2.2.2.6 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 or a closely grouped (e.g., in the same rack) collection of appliances acting as a unit. As such, no intra-TOE communication 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.

2.2.2.7 TOE access

The TOE can be configured to display an informative 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.

2.2.2.8 Trusted path/channels

The TOE protects interactive communication with administrators using SSHv2 for CLI access, for which 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, an 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.

2.3 TOE Documentation

Brocade offers a series of documents that describe the installation of the Brocade MLX and NetIron CER 2000 router products 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:

- Brocade NetIron CES and Brocade NetIron CER Devices Hardware Guide, 53-1002423-02
- Brocade MLX Series and NetIron Family Configuration Guide, 53-1002423-02

- Brocade MLX Series and Brocade NetIron XMR Hardware Installation Guide, 53-1002424-02
- Brocade MLX Series and Brocade NetIron Family YANG Guide, 53-1002427-02
- Brocade MLX Series and Brocade NetIron Family FIPS Guide, 53-1002423-02

3. Security Problem Definition

The Security Problem Definition (composed of organizational policies, threat statements, and assumption) has been drawn from the *Security Requirements for Network Devices*, version 1.1, 8 June 2012 (NDPP). The NDPP offers additional information about the identified threats, 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 Problem Definition appropriate for network infrastructure devices and as such is applicable to the Brocade MLX and NetIron CER 2000 router Family TOE.

3.1 Organizational Policies

P.ACCESS_BANNER

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

3.2 Threats

T.ADMIN_ERROR An administrator may unintentionally install or configure the TOE incorrectly, resulting in ineffective security mechanisms.

T.TSF_FAILURE Security mechanisms of the TOE may fail, leading to a compromise of the TSF.

T.UNAUTHORIZED_ACCESS

A user may gain unauthorized access to the TOE data and TOE executable code. A malicious user, process, or external IT entity may masquerade as an authorized entity in order to gain unauthorized access to data or TOE resources. A malicious user, process, or external IT entity may misrepresent itself as the TOE to obtain identification and authentication data.

T.UNAUTHORIZED_UPDATE

A malicious party attempts to supply the end user with an update to the product that may compromise the security features of the TOE.

T.UNDETECTED_ACTIONS

Malicious remote users or external IT entities may take actions that adversely affect the security of the TOE. These actions may remain undetected and thus their effects cannot be effectively mitigated.

T.USER_DATA_REUSE

User data may be inadvertently sent to a destination not intended by the original sender.

3.3 Assumptions

A.NO_GENERAL_PURPOSE

It is assumed that 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.

A.PHYSICAL

Physical security, commensurate with the value of the TOE and the data it contains, is assumed to be provided by the environment.

A.TRUSTED_ADMIN

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

4. Security Objectives

Like the Security Problem Definition, the Security Objectives have been drawn from the *Security Requirements for Network Devices*, version 1.1, 8 June 2012 (NDPP). 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 MLX and NetIron CER 2000 router Family TOE.

4.1 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.

4.2 Security Objectives for the Environment

OE.NO_GENERAL_PURPOSE

There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE.

OE.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.

5. IT 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): *Security Requirements for Network Devices*, Version 1.1, 8 June 2012 (NDPP). 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 Extended Requirement Definitions

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_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

5.2 TOE Security Functional Requirements

The following table identifies the SFRs that are satisfied by the Brocade MLX and NetIron CER 2000 router Family 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_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

5.2.1 Security audit (FAU)

5.2.1.1 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.

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 1 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. 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:
- Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and
 - 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_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.	

Requirement	Auditable Events	Additional Audit Record Contents
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).
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

5.2.1.2 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.

5.2.1.3 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.

5.2.2 Cryptographic support (FCS)

5.2.2.1 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 "The RSA Validation System (RSA2VS)" 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.

In order to show that the TSF complies with 800-56A and/or 800-56B, depending on the selections made, the evaluator shall ensure that the TSS contains the following information:

- The TSS shall list all sections of the appropriate 800-56 standard(s) to which the TOE complies.
- For each applicable section listed in the TSS, for all statements that are not "shall" (that is, "shall not", "should", and "should not"), if the TOE implements such options it shall be described in the TSS. If the included functionality is indicated as "shall not" or "should not" in the standard, the TSS shall provide a rationale for why this will not adversely affect the security policy implemented by the TOE;
- For each applicable section of 800-56A and 800-56B (as selected), any omission of functionality related to "shall" or "should" statements shall be described;

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.

5.2.2.2 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 key; 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").

5.2.2.3 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, 256-bits, and [none] 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.

5.2.2.4 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" (DSAVS or DSA2VS), "The Elliptic Curve Digital Signature Algorithm Validation System" (ECDSAVS or ECDSA2VS), and "The RSA Validation System" (RSAVS) 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.

5.2.2.5 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-256*] and message digest sizes [*160*, *256*, 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.

5.2.2.6 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]*, key size [*512*], 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.

5.2.2.7 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 [Hash_DRBG (any)]*] seeded by an entropy source that accumulated entropy from [*a software-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 bit length of the keys and authorization factors 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 \leq 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.

5.2.2.8 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, and 4254.

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 a protocol 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).

FCS_SSH_EXT.1.7

The TSF shall ensure that diffie-hellman-group14-sha1 is 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. 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. The evaluator shall then attempt to perform a diffie-hellman-group14-sha1 key exchange, and observe that the attempt succeeds.

5.2.2.9 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)*] supporting the following ciphersuites:

Mandatory Ciphersuites:

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

Optional Ciphersuites:

[*none*].

Assurance Activity:

The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that optional characteristics (e.g., extensions supported, client authentication supported) are specified, and the ciphersuites supported are specified as well. 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 (on the wire) 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).

5.2.3 User data protection (FDP)

5.2.3.1 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.

5.2.4 Identification and authentication (FIA)

5.2.4.1 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: [“!”, “@”, “#”, “\$”, “%”, “^”, “&”, “*”, “(”, “)”, [“_”, “+”, “=”, “-”, “.”, “/”, “:”, “;”, “<”, “=”, “>”, “[”, “\”, “]”, “_”, “^”, “p”, “q”, and “~”];
2. Minimum password length shall be 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.

5.2.4.2 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.

5.2.4.3 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.

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.

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

FIA_UAU_EXT.2.1

The TSF shall provide a local password-based authentication mechanism, [*Public Key Authentication for SSH*] to perform administrative user authentication.

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.

5.2.5 Security management (FMT)

5.2.5.1 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.

5.2.5.2 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*].

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.

5.2.5.3 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

- Authorized Administrator role shall be able to administer the TOE locally;
 - Authorized Administrator role shall be able to administer the TOE remotely;
- are satisfied.

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; then both methods of administration must be exercised during the evaluation team's test activities.

5.2.6 Protection of the TSF (FPT)

5.2.6.1 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. If these values are not stored in plaintext, the TSS shall describe how they are protected/obscured.

5.2.6.2 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.

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.

5.2.6.3 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 cryptographic protocols for establishing a connection with the NTP server, the evaluator shall perform this test using each supported protocol.

5.2.6.4 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.

5.2.6.5 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.

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.

5.2.7 TOE access (FTA)

5.2.7.1 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.

5.2.7.2 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.

5.2.7.3 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.

5.2.7.4 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.

5.2.8 Trusted path/channels (FTP)

5.2.8.1 Trusted Channel (FTP_ITC.1)

FTP_ITC.1.1

Refinement: The TSF shall use [*TLS*] to provide a trusted communication channel between itself and authorized IT entities supporting the following capabilities: audit server, [*no other capabilities*] 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**].

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 is not sent in plaintext.

Test 4: The evaluator shall ensure, for each communication channel with an authorized IT entity, modification of the channel data is detected by the TOE.

Test 5: 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.

5.2.8.2 Trusted Path (FTP_TRP.1)

FTP_TRP.1.1

Refinement: The TSF shall use [SSH] 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.

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 is not sent in plaintext.

Test 4: The evaluator shall ensure, for each method of remote administration, modification of the channel data is detected by the TOE.

Further assurance activities are associated with the specific protocols.

5.3 TOE Security Assurance Requirements

The SARs for the TOE are the EAL 1 components as specified in Part 3 of the Common Criteria (with the exception of some name changes in accordance with the NDPP). 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

5.3.1 Development (ADV)

5.3.1.1 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.

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 there is insufficient interface information, then an adequate functional specification has not been provided.

5.3.2 Guidance documents (AGD)

5.3.2.1 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 fulfil 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.

Assurance Activity:

Some of the contents of the operational guidance will be verified by the assurance activities above 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.

5.3.2.2 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.

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.

5.3.3 Life-cycle support (ALC)

5.3.3.1 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.

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.

5.3.3.2 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.

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.

5.3.4 Tests (ATE)

5.3.4.1 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.

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, 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.

5.3.5 Vulnerability assessment (AVA)

5.3.5.1 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.

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.

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; establishment and termination of a trusted channel or trusted path; user logoff and session termination; changes to the system time; and initiation of a trusted update. 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 them (for which tools accessible via the CLI 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 a configurable quantity of audit logs that can be stored locally, 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 send a copy of the audit records if so desired. The TOE can be configured to use TLS to protect audit logs exported to an external server. When an audit record is generated, the TOE stores the record locally and also sends that record to the selected SYSLOG server. If the active SYSLOG server is down, the TOE establishes a TLS connection to the next configured SYSLOG server. During this time, when there is no active SYSLOG server, the generated audit logs are queued for up to 60 seconds hold-down time for later offload. At the end of the hold-down time, the TOE checks for any active SYSLOG server over TLS and sends all the queued audit logs to that server.

The TOE acts as a TLS client when it uses TLS to connect to an external syslog server. As such, it requires a client certificate and private key in order for the secure syslog server connection to be configured. Once established, the channel stays up indefinitely, regardless of inactivity on the channel. It closes if the external syslog server closes the channel or an administrator deletes the active connection on the TOE. When the TOE discovers the session is disconnected, it will attempt to connect to the next configured syslog server, and queued messages (during a hold-down time) are sent when the TLS channel is up again.

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 evaluated configuration requires that the TOE operates in FIPS Common Criteria mode.

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

Functions	Standards	Certificates
Asymmetric key generation		
<ul style="list-style-type: none"> • Domain parameter generation and key establishment¹ 	NIST Special Publication 800-56B	1217
Encryption/Decryption		
<ul style="list-style-type: none"> • AES and CBC (128, or 256 bits) 	FIPS PUB 197 NIST SP 800-38A	2359
Cryptographic signature services		
<ul style="list-style-type: none"> • RSA Digital Signature Algorithm (rDSA) (modulus 2048) 	FIPS PUB 186-2	1217
Cryptographic hashing		
<ul style="list-style-type: none"> • SHA-1, and SHA-256 (digest sizes 160, 256) 	FIPS Pub 180-3	2031
Keyed-hash message authentication		
<ul style="list-style-type: none"> • HMAC-SHA-1 (digest size 160) 	FIPS Pub 198-1 FIPS Pub 180-3	1462
Random bit generation		
<ul style="list-style-type: none"> • RGB with software with a minimum of 256 bits of non-determinism 	NIST Special Publication 800-90 using Hash_DRBG	301

Table 4 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

¹ Note that 2048-bit RSA keys are estimated to have key strength roughly equivalent to a symmetric key strength of 112 bits

NIST SP800-56B Section Reference	“should”, “should not”, or “shall not”	Implemented?	Rationale for deviation
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
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 5 NIST SP800-56B Conformance

The TOE uses a software-based random bit generator that complies with Special Publication 800-90 using Hash_DRBG (any) when operating in the FIPS mode. SHA-256 is used in conjunction with a minimum of 440 bits of entropy accumulated from the processing stack, hardware serial numbers, and the low-order bits from the current time of day.

Additionally, the TOE is designed to zeroize secret and private keys when they are no longer required by the TOE. 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 SSH session key is transient. It is zeroized at the end of a session and recreated at the beginning of a new session. The DRBG seed is recomputed periodically on 100 millisecond intervals. Each time this occurs, four bytes of the seed are written into an 8K buffer. When the buffer is full the DRBG V and C values are regenerated. 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. The

SSH session key is transient. It is stored in RAM and zeroized by overwriting once with zeroes at the end of a session.

These supporting cryptographic functions are included to support the SSHv2 (compliant with RFCs 4251, 4252, 4253, and 4254).

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 FIPS mode.

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. 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 by overwriting once with zeroes.

Executing the no fips enable command zeroizes all RSA private keys.”

The DRBG seed and Hash DRBG Entropy are stored in RAM and recomputed periodically on 100 millisecond intervals. Each time this occurs, four bytes of the seed are written into an 8K buffer. When the buffer is full the DRBG V and C values are regenerated-the new value overwrites the old value.

The DH private exponent is stored in RAM and 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.

The TOE implements TLSv1.0 in accordance with RFC 2246. The TOE’s implementation supports the following ciphersuites:

- 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

The TOE supports TLS sessions for exporting audit data. The TLS pre-master secret is generated during the TLS handshake. It is stored in RAM and overwritten once with zeroes after the TLS handshake is complete. The TLS session key is generated for every TLS session. It is stored in RAM and overwritten once with zeroes after the session is closed.

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

- FCS_CKM.1: See table above.
- FCS_CKM_EXT.4: Keys are zeroized when they are no longer needed by the TOE.
- FCS_COP.1(1): See table above.
- FCS_COP.1(2): See table above.
- FCS_COP.1(3): See table above.
- FCS_COP.1(4): See table above.
- FCS_RBG_EXT.1: See table 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 sessions for exporting audit data.

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 a network packet is sent, the buffer used by the packet is recalled and managed by the buffer pool. After that, if a new packet acquires a buffer from the buffer pool, the new packet data will be used to overwrite any previous data in the buffer. If an allocated buffer exceeds the size of the packet, and additional space will be overwritten (padded) with zeros.

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 warning banners and to permit network traffic to flow through the TOE without identification or authentication so long as it conforms to the information flow policy rules. The TOE authenticates TOE Users against their user name, and password. Passwords for privilege levels cannot be configured and user accounts must be created with privilege levels (0- super, 4-port, or 5-read-only). Administrators can access the TOE remotely via SSH. The TOE supports both password and public key-based authentication mechanisms over SSH.

The Authorized Administrator with Super User privilege is able to define 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.

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). Alternative 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 and public key for the Super User Privilege level, Local User Accounts configured on the device. When authentication is successful, the TOE is provided the applicable privilege defined for the associated user.

The Authentication Method List is ordered so that it will be processed from first to last. In each case, the user authentication will succeed, fail, or result in an error. Only in the case of an error (e.g., an external server is unavailable) will processing proceed to the next authentication method in the list. If a given authentication method succeeds, the user will be logged in and will be able to perform functions according to their privilege level. If a given authentication mechanism fails, the user will be denied a login session. If the point is reached where every authentication method on the list fails, only an authorized administrator with Super User privilege whose Super User password is not rejected will succeed in logging in to the system. If all of the authorized administrator's Super User passwords are rejected only the locally defined Super User can login.

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 some interfaces provide feedback as '*' while others provide no feedback (the password prompt remains empty) as you enter passwords.
- FIA_UAU_EXT.2: The TOE can be configured to utilize local password-based authentication.
- 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 warning 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). 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, 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 function, as well as the non-security relevant aspects of the TOE.

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 using digital signatures.
- 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. The TOE operates in FIPS Common Criteria mode.

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 does not allow reading of private keys nor it allows access to any user to read or write to any memory location. The public keys are encrypted using base64 while in the compact flash.

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. 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 also uses the clock for measuring session inactivity.

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. The built-in self tests include basic read-write memory, flash read, software checksum tests, and device detection tests. The memory is tested by writing a value to each memory location and reading it back to ensure it is identical to what was written. Furthermore, the TOE is designed to query each pluggable module which in turn includes its own diagnostics that will serve to help identify any failing modules. When operating in FIPS mode, the power-on self-tests comply with the CAVP requirements for self-testing. At power up the TOE performs cryptographic algorithm testing which includes testing against RC2-40bit, RC4-40bit, DES-56bit, TDES-56 bit, AES (128,192,256-bit), MD2,MD5, SHA(1,256,384,512 bit), HMAC-SHA(1,256,384,512 bit),RSA 2048, DSA 1024, and DRBG known answer tests. The TOE also performs firmware integrity and firmware load tests (DSA 1024 bit, SHA1 Signature Verification) during power up. Additional power-up testing includes continuous random number generator tests, RSA1024/2048 SHA-1 pairwise consistency tests, bypass tests and manual key entry tests.

The TOE supports loading a new software image either automatically (when so configured) or manually by the administrator using CLI commands. When automatic updates are configured, each time the TOE boots it will attempt to connect to a configured TFTP server in order to compare the current software image name to those available. If a new software version is available it will be automatically downloaded to the TOE. From the CLI, an administrator can use either TFTP or SCP in order to download a software image. In either case, prior to actually installing and using the new software image, its digital certificate is verified using the public key in the certificate configured in the TOE. The TOE uses a RSA digital Signature (modulus 2048) Algorithm. The updates are signed by Brocade and the TOE verifies the images through use of an applicable Brocade public certificate which comes preinstalled on the TOE. An unverified image cannot be installed.

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 a login banner. The login banner can be configured to display welcome information in conjunction with login prompts. There is a size limit to the banner text of 2 Kbytes. 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. After a console session is terminated, a new screen is displayed that shows the banner and a text asking for login credentials.

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 to provide a trusted path for administrators remotely accessing the TOE's CLI. 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. Furthermore, the TOE maintains a list of users that are allowed to access the TOE remotely. As such, even when a session can be negotiated, the TOE then checks to ensure the user is authorized for remote administration and if not the session is dropped.

When a client attempts to connect using SSH, the TOE and the client will negotiate the most secure algorithms available at both ends to protect that session.

In each case, AES-CBC with 256-, or 128-bit keys is implemented for encryption and decryption and RSA using up to 2048-bit keys are implemented for key exchange and authentication. DSA is disabled in the evaluated configuration.

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

The TOE implements TLSv1.0 (see Section 6.2) to provide a trusted channel between itself and external SYSLOG servers.

The TOE update service is secured using SCP. TFTP is disabled in FIPS CC mode and so cannot be used in the evaluated configuration..

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 exported audit records are sent only to the configured server so they are not subject to inappropriate disclosure or modification. The TOE update service can be secured using SCP.
- FTP_TRP.1: The TOE provides SSH, 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 CAVP certified cryptographic operations, and all remote security management functions require the use of one of these secure channels.

7. Protection Profile Claims

The ST conforms to the *Security Requirements for Network Devices*, version 1.1, 8 June 2012 (NDPP). As explained previously, the security problem definition, security objectives, and security requirements have been drawn from the NDPP.

8. Rationale

This section provides the rationale for completeness and consistency of the Security Target. The rationale addresses the following areas:

- Security Objectives;
- Security Functional Requirements;
- Security Assurance Requirements;
- Requirement Dependencies;
- TOE Summary Specification.

8.1 Security Objectives Rationale

This section shows that all secure usage assumptions, organizational security policies, and threats are completely covered by security objectives. In addition, each objective counters or addresses at least one assumption, organizational security policy, or threat.

8.1.1 Security Objectives Rationale for the TOE and Environment

This section provides evidence demonstrating the coverage of organizational policies and usage assumptions by the security objectives.

	P.ACCESS_BANNER	T.ADMIN_ERROR	T.TSF_FAILURE	T.UNAUTHORIZED_ACCESS	T.UNAUTHORIZED_UPDATE	T.UNDETECTED_ACTIONS	T.USER_DATA_REUSE	A.NO_GENERAL_PURPOSE	A.PHYSICAL	A.TRUSTED_ADMIN
O.DISPLAY_BANNER	X									
O.PROTECTED_COMMUNICATIONS				X						
O.RESIDUAL_INFORMATION_CLEARING							X			
O.SESSION_LOCK				X						
O.SYSTEM_MONITORING		X		X		X				
O.TOE_ADMINISTRATION				X						
O.TSF_SELF_TEST			X							
O.VERIFIABLE_UPDATES					X					
OE.NO_GENERAL_PURPOSE								X		
OE.PHYSICAL									X	
OE.TRUSTED_ADMIN										X

Table 6 Environment to Objective Correspondence

8.1.1.1 P.ACCESS_BANNER

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

This Organizational Policy is satisfied by ensuring that:

- O.DISPLAY_BANNER: To fulfill the policy to display advisory information to users prior to their use of the TOE, the TOE is expected to display a configured banner when users login to establish an interactive session.

8.1.1.2 T.ADMIN_ERROR

An administrator may unintentionally install or configure the TOE incorrectly, resulting in ineffective security mechanisms.

This Threat is satisfied by ensuring that:

- O.SYSTEM_MONITORING: To reduce the potential of an administrative error that might be unnoticed or untraceable, the TOE is expected to log security relevant events and export those logs to an external log server.

8.1.1.3 T.TSF_FAILURE

Security mechanisms of the TOE may fail, leading to a compromise of the TSF.

This Threat is satisfied by ensuring that:

- O.TSF_SELF_TEST: To reduce the potential for undetected TOE failures and to help ensure that the TOE security functions are operating properly, the TOE is expected to perform self-tests.

8.1.1.4 T.UNAUTHORIZED_ACCESS

A user may gain unauthorized access to the TOE data and TOE executable code. A malicious user, process, or external IT entity may masquerade as an authorized entity in order to gain unauthorized access to data or TOE resources. A malicious user, process, or external IT entity may misrepresent itself as the TOE to obtain identification and authentication data.

This Threat is satisfied by ensuring that:

- O.PROTECTED_COMMUNICATIONS: To reduce the potential that an attacker might gain unauthorized access to the TOE or its data via data transmitted across a network, the TOE is expected to protect its communication channels.
- O.SESSION_LOCK: To reduce the potential for unauthorized access to TOE security functions and data, the TOE is expected to lock or terminate unattended or inactive sessions.
- O.SYSTEM_MONITORING: To reduce the potential of unauthorized access attempts that might go unnoticed, the TOE is expected to log security relevant events and export those logs to an external log server.
- O.TOE_ADMINISTRATION: To reduce the potential of unauthorized access to TOE security functions and data, the TOE is expected to be designed to ensure that only presumably authorized administrators can log in and access security management functions.

8.1.1.5 T.UNAUTHORIZED_UPDATE

A malicious party attempts to supply the end user with an update to the product that may compromise the security features of the TOE.

This Threat is satisfied by ensuring that:

- O.VERIFIABLE_UPDATES: To reduce the potential that an update might contain malicious or unintended features, the TOE is expected to provide mechanisms that serve to ensure the integrity of updates prior to their use.

8.1.1.6 T.UNDETECTED_ACTIONS

Malicious remote users or external IT entities may take actions that adversely affect the security of the TOE. These actions may remain undetected and thus their effects cannot be effectively mitigated.

This Threat is satisfied by ensuring that:

- O.SYSTEM_MONITORING: To reduce the potential of security relevant actions occurring without notice, the TOE is expected to log security relevant events and export those logs to an external log server.

8.1.1.7 T.USER_DATA_REUSE

User data may be inadvertently sent to a destination not intended by the original sender.

This Threat is satisfied by ensuring that:

- O.RESIDUAL_INFORMATION_CLEARING: To reduce the potential of data being erroneously sent to an unintended recipient, the TOE is expected to ensure that residual data is appropriately managed.

8.1.1.8 A.NO_GENERAL_PURPOSE

It is assumed that 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.

This Assumption is satisfied by ensuring that:

- 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.

8.1.1.9 A.PHYSICAL

Physical security, commensurate with the value of the TOE and the data it contains, is assumed to be provided by the environment.

This Assumption is satisfied by ensuring that:

- OE.PHYSICAL: Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.

8.1.1.10 A.TRUSTED_ADMIN

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

This Assumption is satisfied by ensuring that:

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

8.2 Security Requirements Rationale

This section provides evidence supporting the internal consistency and completeness of the components (requirements) in the Security Target. Note that **Table 7** indicates the requirements that effectively satisfy the individual objectives. .

8.2.1 Security Functional Requirements Rationale

All Security Functional Requirements (SFR) identified in this Security Target are fully addressed in this section and each SFR is mapped to the objective for which it is intended to satisfy.

	O.DISPLAY_BANNER	O.PROTECTED_COMMUNICATIONS	O.RESIDUAL_INFORMATION_CLEARING	O.SESSION_LOCK	O.SYSTEM_MONITORING	O.TOE_ADMINISTRATION	O.TSF_SELF_TEST	O.VERIFIABLE_UPDATES
FAU_GEN.1					X			
FAU_GEN.2					X			
FAU_STG_EXT.1					X			
FCS_CKM.1		X						
FCS_CKM_EXT.4		X						
FCS_COP.1(1)		X						
FCS_COP.1(2)		X						X
FCS_COP.1(3)		X						X
FCS_COP.1(4)		X						
FCS_RBG_EXT.1		X						
FCS_SSH_EXT.1		X						
FCS_TLS_EXT.1		X						
FDP_RIP.2			X					
FIA_PMG_EXT.1						X		
FIA_UAU.7						X		
FIA_UIA_EXT.1						X		
FIA_UAU_EXT.2						X		
FMT_MTD.1						X		
FMT_SMF.1						X		
FMT_SMR.2						X		
FPT_SKP_EXT.1		X						
FPT_APW_EXT.1						X		
FPT_STM.1					X			
FPT_TST_EXT.1							X	
FPT_TUD_EXT.1								X
FTA_SSL.3				X		X		
FTA_SSL.4						X		
FTA_SSL_EXT.1				X		X		
FTA_TAB.1	X							
FTP_ITC.1		X						
FTP_TRP.1		X						

Table 7 Objective to Requirement Correspondence

8.2.1.1 O.DISPLAY_BANNER

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

This TOE Security Objective is satisfied by ensuring that:

- FTA_TAB.1: The TOE is required to display the configured advisory banner whenever a user/administrator connects to the TOE.

8.2.1.2 O.PROTECTED_COMMUNICATIONS

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

This TOE Security Objective is satisfied by ensuring that:

- FCS_CKM.1: The TOE is required to be able to generate encryption keys to support other cryptographic operations.
- FCS_CKM_EXT.4: The TOE is required to zeroize keys when no longer need to prevent subsequent disclosure.
- FCS_COP.1(1): The TOE is required to implement FIPS-conformant AES in support of cryptographic protocols.
- FCS_COP.1(2): The TOE is required to implement FIPS-conformant rDSA in support of cryptographic protocols.+
- FCS_COP.1(3): The TOE is required to implement FIPS-conformant SHA-1, SHA-256, SHA-384, and/or SHA-512 in support of cryptographic protocols.
- FCS_COP.1(4): The TOE is required to implement FIPS-conformant HMAC SHA-1 in support of cryptographic protocols.
- FCS_RBG_EXT.1: The TOE is required to implement NIST- or FIPS-conformant Random Bit Generation in support of cryptographic protocols.
- FCS_SSH_EXT.1: The TOE is required to implement SSH properly to protect applicable network communication channels.
- FCS_TLS_EXT.1: The TOE is required to implement TLS properly to protect applicable network communication channels.
- FPT_SKP_EXT.1: The TOE is required to prevent even administrators from readily accessing sensitive user and TSF data such as cryptographic keys.
- FTP_ITC.1: The TOE is required to protect communication between itself and its external peers from disclosure and modification.
- FTP_TRP.1: The TOE is required to protect communication between itself and its administrators from disclosure and modification.

8.2.1.3 O.RESIDUAL_INFORMATION_CLEARING

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

This TOE Security Objective is satisfied by ensuring that:

- FDP_RIP.2: The TOE is required to clear all information when allocating storage resources for subsequent activities.

8.2.1.4 O.SESSION_LOCK

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

This TOE Security Objective is satisfied by ensuring that:

- FTA_SSL.3: The TOE is required to terminate remote sessions after an administrator defined period of inactivity indicating the user may not be in attendance.
- FTA_SSL_EXT.1: The TOE is required to lock or terminate local sessions after an administrator defined period of inactivity indicating the user may not be in attendance.

8.2.1.5 O.SYSTEM_MONITORING

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

This TOE Security Objective is satisfied by ensuring that:

- FAU_GEN.1: The TOE is required to be able to generate audit events for security relevant activities on the TOE.
- FAU_GEN.2: The TOE is required to associate audit events to users to ensure proper accountability.
- FAU_STG_EXT.1: The TOE is required to be able to export audit records to an external audit server via a secure channel to protect the integrity and security of those records.
- FPT_STM.1: The TOE is required to generate reliable time stamps to be used in its audit records for proper accounting.

8.2.1.6 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.

This TOE Security Objective is satisfied by ensuring that:

- FIA_PMG_EXT.1: The TOE is required to implement mechanisms allowing an administrator to constrain the construction of passwords to encourage more secure (or harder to guess) passwords.
- FIA_UAU.7: The TOE is required to not echo passwords when being entered to mitigate the chance of an accidental password disclosure.
- FIA_UIA_EXT.1: The TOE is required to ensure that users must be identified and authenticated in order to access functions, other than those specifically intended to be accessed without identification and authentication.
- FIA_UAU_EXT.2: The TOE is required to implement a local authentication mechanism and can support additional authentication mechanisms.
- FMT_MTD.1: The TOE is required to restrict access to security relevant data to administrators.
- FMT_SMF.1: The TOE is required to provide a minimum set of security functions to ensure the TOE security features can be properly managed.
- FMT_SMR.2: The TOE is required to implement a minimum of an Authorized Administrator role and can implement additional roles where necessary.
- FPT_APW_EXT.1: The TOE is required to prevent even administrators from readily accessing sensitive user and TSF data such as passwords.
- FTA_SSL.3: The TOE is required to terminate remote sessions after an administrator defined period of inactivity indicating the administrator may not be in attendance.
- FTA_SSL.4: The TOE allows users to terminate their sessions at any time to help them ensure their credentials are not inappropriately used.
- FTA_SSL_EXT.1: The TOE is required to lock or terminate local sessions after an administrator defined period of inactivity indicating the administrator may not be in attendance.

8.2.1.7 O.TSF_SELF_TEST

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

This TOE Security Objective is satisfied by ensuring that:

- FPT_TST_EXT.1: The TOE is required to exercise self-tests during start-up to periodically ensure that the TOE security functions appear to be operating correctly.

8.2.1.8 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.

This TOE Security Objective is satisfied by ensuring that:

- FCS_COP.1(2): The TOE is required to either use digital signatures or cryptographic hashes to ensure the integrity of updates.
- FCS_COP.1(3): The TOE is required to either use digital signatures or cryptographic hashes to ensure the integrity of updates.
- FPT_TUD_EXT.1: The TOE is required to provide update functions and also the means for an administrator to initiate and verify updates before they are applied.

8.3 Security Assurance Requirements Rationale

The Security Assurance Requirements (SARs), which correspond to EAL1, in this ST represent the SARs identified in the NDPP.

Note that the NDPP includes a number of ‘Assurance Activities’ which are in effect refinements of the underlying SARs. As such, those assurance activities have been reproduced in this ST since they need be addressed in the context of the evaluation.

8.4 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_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

ST Requirement	CC Dependencies	ST Dependencies
FIA_PMG_EXT.1	none	none
FIA_UAU.7	FIA_UAU.1	FIA_UIA_EXT.1
FIA_UIA_EXT.1	none	none
FIA_UAU_EXT.2	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_SKP_EXT.1	none	none
FPT_APW_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 8 Requirement Dependencies

8.5 TOE Summary Specification Rationale

Each subsection in Section 6, the TOE Summary Specification, describes a security function of the TOE. Each description is followed with rationale that indicates which requirements are satisfied by aspects of the corresponding security function. The set of security functions work together to satisfy all of the security functions and assurance requirements. Furthermore, all of the security functions are necessary in order for the TSF to provide the required security functionality.

This Section in conjunction with Section 6, the TOE Summary Specification, provides evidence that the security functions are suitable to meet the TOE security requirements. The collection of security functions work together to provide all of the security requirements. The security functions described in the TOE summary specification are all necessary for the required security functionality in the TSF. **Table 9 Security Functions vs. Requirements Mapping** demonstrates the relationship between security requirements and security functions.

	Security audit	Cryptographic support	User data protection	Identification and authentication	Security management	Protection of the TSF	TOE access	Trusted path/channels
FAU_GEN.1	X							
FAU_GEN.2	X							
FAU_STG_EXT.1	X							
FCS_CKM.1		X						
FCS_CKM_EXT.4		X						
FCS_COP.1(1)		X						
FCS_COP.1(2)		X						
FCS_COP.1(3)		X						
FCS_COP.1(4)		X						
FCS_RBG_EXT.1		X						
FCS_SSH_EXT.1		X						
FCS_TLS_EXT.1		X						
FDP_RIP.2			X					
FIA_PMG_EXT.1				X				
FIA_UAU.7				X				
FIA_UIA_EXT.1				X				
FIA_UAU_EXT.2				X				
FMT_MTD.1					X			
FMT_SMF.1					X			
FMT_SMR.2					X			
FPT_SKP_EXT.1						X		
FPT_APW_EXT.1						X		
FPT_STM.1						X		
FPT_TST_EXT.1						X		
FPT_TUD_EXT.1						X		
FTA_SSL.3							X	
FTA_SSL.4							X	
FTA_SSL_EXT.1							X	
FTA_TAB.1							X	
FTP_ITC.1								X
FTP_TRP.1								X

Table 9 Security Functions vs. Requirements Mapping