Lumeta IPsonar Security Target

Version 1.0
10/07/13

Prepared for:

Lumeta Corporation

300 Atrium Drive, 3rd Floor
Somerset, New Jersey 08873

Prepared By:

Leidos, Incorporated

(formerly Science Applications International Corporation)

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

LIST OF TABLES

1. Security Target Introduction

This section identifies the Security Target (ST) and Target of Evaluation (TOE) identification, ST conventions, ST conformance claims, and the ST organization. The TOE is IPsonar provided by Lumeta Corporation. The IPsonar product is an active network discovery solution for routed and switched network infrastructures. At a high level, it facilitates the discovery of the overall infrastructure of the attached network including identification of network, host, and other devices including potentially unauthorized network connectivity.

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.

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 – Lumeta IPsonar Security Target

ST Version – Version 1.0

ST Date – 10/07/13

TOE Identification – Lumeta IPsonar 5.5C (Report/Scan server version 5.5.0.12174101C; Sensor version 5.5.3.12338104)

TOE Developer – Lumeta Corporation

Evaluation Sponsor – Lumeta Corporation

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 Protection Profile for Network Devices, Information Assurance Directorate, Version 1.1, 08 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
    - Assurance Level: NDPPt

1.3 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.

2. TOE Description

The Target of Evaluation (TOE) is the Lumeta IPsonar 5.5C running on FreeBSD-8.1 operating system. The OS is an integral part of the product and is not managed separately, even for upgrades which Lumeta provides and the OS is upgraded in a controlled way. The TOE can be deployed as a standalone appliance or alternately as a series of cooperating appliances depending on the specific needs of the user.

The TOE is available as either a 1U rack-mountable appliance or alternately as a preconfigured laptop. The hardware is commodity computers running x86 64-bit compatible CPU. The specific hardware tested is Dell d1950 PowerEdge and HP ProBook 6555b. The same security and functional capabilities are available regardless of the physical form factor.

The SFTP feature in IPsonar to upload certain prior saved data is not in the scope of this evaluation.

See http://www.lumeta.com/product/hardware.html for more information about the two hardware options.

2.1 TOE Overview

The TOE is designed to plug into a network and to actively examine and discover the network infrastructure. To that end it can identify and examine network connected assets such as hosts and other network devices in order to create a view of the routed infrastructure associated with the attached network. It primary functions include:

- **Network Discovery** – Identifies all network address spaces, routing devices and connectivity flows across the network (including "stealth" assets, i.e. hidden devices that do not respond to queries) utilizing advanced multi-protocol discovery technology, and creates a comprehensive route-based topology that identifies a network's true perimeter. *Host Topology Visualization / Layer 2:* An optional product module supports layer 2 topology mapping, stealthy device identification, guest network and extranet security, VLAN compliance and Virtual Machine identification. The TOE can be operated with or without this module present.

- **Host Discovery** – Detects all known and previously unknown network devices by conducting a census of IP addresses across protocols. Flags devices unrecognized by official network inventories for remediation.
- **Leak Discovery** – Reveals unauthorized connections between a network and another network, sub-net, or the Internet, and determines whether access is outbound, inbound or both. Leak discovery highlights unknown connections into other organizations (e.g., legacy divestiture connectivity) or to the Internet.
- **Device Discovery** – Identifies web services, wireless access points and IP applications active on hosts and devices – including those not owned by the client or its employees – pinpointing resources for which tested ports are active. Additionally, Layer 2 discovery matches a device's unique MAC address with its assigned IP address, providing crucial information for asset management and diagnostics.

Note, however, that while these are the primary functions of the TOE, the evaluation does not specifically address these capabilities. Rather, the evaluation (and hence this security target) focuses on the security of the device as a network infrastructure component as required in the NDPP. The evaluated functions of the TOE are further explored in the remainder of this Security Target.

2.2 TOE Architecture

The TOE can be deployed as a single stand-alone appliance or as a collection of cooperating appliances. In the latter case some of the appliances are configured to provide only a subset of the overall functions of the IPsonar appliance. The following list summarizes the modes of operation available within each IPsonar appliance:

- **Sensors. N**etwork scanning is achieved through the use of network entry points called Sensors. The TOE can be deployed as a Sensor so that it can collect information about its connected network and forward that information to a configured Scan Server.
- **Scan Servers.** These Scan Servers are positioned at appropriate points in the network to ensure connectivity with any distributed Sensors. Multiple scans can be run simultaneously by using multiple configured Sensors.
- **Report Servers.** Functioning as the data repository, Report Servers separate report generation from scanning to reduce IPsonar's operational footprint. A single remote Report Server can support multiple configured Scan Servers.

A given deployed instance of the IPsonar appliance can be configured to include one or more of the operational modes summarized above. However, a deployed TOE configuration necessarily must include one or more Sensors, one or more Scan Servers, and a Report Server so that all of those capabilities are present among one or more deployed IPsonar appliances.

When the TOE is distributed the communication among the appliances is protected using HTTPS/TLS. Administrative access to any of the servers is similarly protected using HTTPS/TLS.

2.2.1 Physical Boundaries

The TOE consists of one or more physical appliances (identified above). Each appliance can be deployed in one of three main configurations: Network Sensor, Network Sensor/Scan Server, or Network Sensor/Scan Server/Report Server. Each appliance includes physical network connections allowing access to the subject networks, communication among associated appliances, and remote access by administrators.

The TOE can utilize services or components available in the operational environment as follows:

- NTP Server – used to synchronize the internal clock. This is recommended, but not required.
- IF-MAP - The InterFaces for Metadata Access Point is an open standard client/server protocol developed by the Trusted Computing Group (TCG) as one of the core protocols of the Trusted Network Connect (TNC) open architecture. IPsonar can be configured to send IPsonar Alerts to an IF-MAP Server.
- SYSLOG Server - Syslog is used both for normal host and IPsonar activity monitoring, and optionally for IPsonar Alerts, but not for general security auditing of the TOE. It is enabled by default to log locally and

this cannot be changed. It is optional to specify that all logging is additionally sent to a configured external syslogd server.

- LDAP Server - IPsonar can retrieve CRLs (Certificate Revocation List updates) from an LDAP server. This is an optional configuration, and even if certificates are used for user authentication this is still optional. In effect, IPsonar can use a PKI CRL server via LDAP.
- PKI Server – used as a Certificate Authority for certificate generation and to obtain CRLs used to ensure configured certificates have not expired or are not revoked. This is only needed if certificates are configured for authentication of one or more users. Also, access need only be indirect since certificates can be entered by administrators and CRLs can be accessed via LDAP.

  Note that PKI CRL and PKI Server use was not evaluated.
- Management Workstation – used to access the TOE via SSHv2 or HTTPS/TLS in order to remotely administer the TOE.

2.2.2 Logical Boundaries

This section summarizes the security functions provided by IPsonar:

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

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

1. The TOE provides default user accounts for access to the console. These user accounts enable anonymous logins to the TOE. Anonymous logins are not permitted in the evaluated configuration and must be disabled. During configuration, Authorized Administrators must create password-controlled logins for TOE users to access the console. These user accounts replace the pre-configured anonymous accounts. Once the procedures for configuring the new user accounts have been followed, anonymous logins are disabled.
2. The evaluated configuration requires these additional elements:
   a. The TOE must be configured to operate in FIPS mode. This configuration is performed prior to shipment of the TOE to the customer and cannot be changed. The TOE does not offer the ability to change the FIPS mode configuration and there are no commands availability to modify the FIPS settings.
   b. The Authorized Administrator must follow the instructions provided in Appendix A of the IPsonar Administrator Guide (IP_55CC_AG.docx) for configuring the TOE into the evaluated configuration. This includes performing additional settings such as enabling the Secure System mode by running the *secure_system.sh* command; enabling the password controls; and creating the replacement users as described above.

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 uses FreeBSD-based auditing features that can be configured to store the logs locally so they can be accessed by an administrator and also sent to a remote log server using syslog-ng in order to protect the exported records using TLS.

2.2.2.2 Cryptographic support

The TOE includes the OpenSSL FIPS Object Module that provides key management, random bit generation, encryption/decryption, digital signature and secure hashing features in support of higher level cryptographic protocols including SSH and TLS/HTTPS. Cryptographic algorithms provided by OpenSSL FIPS Object Module have been tested on IPsonar 5.5C under the Cryptographic Algorithm Validation Program (CAVP).

[1]Footnote

2.2.2.3 User data protection

The TOE performs a variety of network infrastructure detection functions, but as a rule does not pass data among network entities. The exception is that data is passed among distributed TOE appliances. Otherwise, it collects data from the network and attached components and ultimately forwards information to TOE administrators.

Regardless, the TOE is designed to ensure that memory and other storage resources are reused properly to mitigate potential data corruption or repetition.

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. It provides the ability to both assign attributes (user names, passwords and roles/privilege levels) and to authenticate users against these attributes. Users can optionally be configured with public certificates so that public/private key authentication can be used.

2.2.2.5 Security management

The TOE provides menu-driven console (Console) commands and a Web-based Graphical User Interface (Web GUI) to access the wide range of security management functions to manage its security policies. 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.

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 private 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 an associated collection of appliances acting together. The communication between associated appliances is protected using TLS.

The TOE includes functions to perform self-tests so that it might detect when it is failing. It also includes mechanisms so that the TOE itself can be updated while ensuring that the updates will not introduce malicious or other unexpected changes in the TOE.

[1] Lumeta ported the OpenSSL FIPS Object Module to FreeBSD 8.1 in IPsonar in accordance with the FIPS Security Policy (http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/1401val2008.htm#1051). Please see section "G.5 Maintaining validation compliance of software or firmware cryptographic modules" in FIPS 140-2 Implementation Guidance regarding porting (http://csrc.nist.gov/groups/STM/cmvp/standards.html).

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 Console access or TLS/HTTPS for Web graphical user interface access. In each case, the both integrity and disclosure protection is ensured. . If the negotiation of an encrypted session fails or if the user does not have authorization for remote administration, an attempted connection will not be established.

The TOE protects communication with an audit log server using TLS connections as part of a syslog-ng implementation to prevent unintended disclosure or modification of logs.

2.3 TOE Documentation

Lumeta offers a document that describes the installation of IPsonar as well as guidance for subsequent use and administration of the applicable security features. The following documents were examined as part of the evaluation:

- IPsonar Administrator Guide
- IPsonar Access Accountability Managing Console Login Accounts
- Format of Syslog Events in IPsonar
- IPsonar Common Criteria Guide

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 1.99, 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 IPsonar 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 Protection Profile for Network Devices, Information Assurance Directorate, Version 1.1, 08 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 IPsonar 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 1.990, 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 NDPP 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 NDPP 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_HTTPS_EXT.1: Explicit: HTTPS
- FCS_RBG_EXT.1: Extended: Cryptographic Operation (Random Bit Generation)
- FCS_SSH_EXT.1: Explicit: SSH
- FCS_TLS_EXT.1: Explicit: TLS
- FIA_PMG_EXT.1: Password Management
- FIA_UAU_EXT.2: Extended: Password-based Authentication Mechanism
- FIA_UIA_EXT.1: User Identification and Authentication
- 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 IPsonar TOE.

Requirement Class	Requirement Component
FAU: Security audit	FAU_GEN.1: Audit Data Generation
	FAU_GEN.2: User identity association

Requirement Class	Requirement Component
	FAU_STG_EXT.1: External Audit Trail Storage
FCS: Cryptographic support	FCS_CKM.1: Cryptographic Key Generation (for asymmetric keys)
	FCS_CKM_EXT.4: Cryptographic Key Zeroization
	FCS_COP.1(1): Cryptographic Operation (for data encryption/decryption)
	FCS_COP.1(2): Cryptographic Operation (for cryptographic signature)
	FCS_COP.1(3): Cryptographic Operation (for cryptographic hashing)
	FCS_COP.1(4): Cryptographic Operation (for keyed-hash message authentication)
	FCS_HTTPS_EXT.1: Explicit: HTTPS
	FCS_RBG_EXT.1: Extended: Cryptographic Operation (Random Bit Generation)
	FCS_SSH_EXT.1: Explicit: SSH
	FCS_TLS_EXT.1: Explicit: TLS
FDP: User data protection	FDP_RIP.2: Full Residual Information Protection
FIA: Identification and authentication	FIA_PMG_EXT.1: Password Management
	FIA_UAU.7: Protected Authentication Feedback
	FIA_UAU_EXT.2: Extended: Password-based Authentication Mechanism
	FIA_UIA_EXT.1: User Identification and Authentication
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_ITT.1: Basic Internal TSF Data Transfer Protection
	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**.

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.	

Requirement	Auditable Events	Additional Audit Record Contents
FCS_COP.1(1)	None.	
FCS_COP.1(2)	None.	
FCS_COP.1(3)	None.	
FCS_COP.1(4)	None.	
FCS_HTTPS_EXT.1	Failure to establish a HTTPS Session. Establishment/Termination of a HTTPS session.	Reason for failure. Non-TOE endpoint of connection (IP address) for both successes and failures.
FCS_RBG_EXT.1	None.	
FCS_SSH_EXT.1	Failure to establish an SSH session. Establishment/Termination of an SSH session.	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.	Reason for failure. Non-TOE endpoint of connection (IP address) for both successes and failures.
FDP_RIP.2	None.	
FIA_PMG_EXT.1	None.	
FIA_UAU_EXT.2	All use of the authentication mechanism.	Origin of the attempt (e.g., IP address).
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.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_ITT.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

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_GEN1.2, and the additional information specified in Table 1 (of the NDPP).

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. If HTTPS is implemented, the test demonstrating the establishment and termination of a TLS session can be combined with the test for an HTTPS session. For administrative actions, the evaluator shall test that each action determined by the evaluator above to be security relevant in the context of the NDPP is auditable. When verifying the test results, the evaluator shall ensure the audit records generated during testing match the format specified in the administrative guide, and that the fields in each audit record have the proper entries.

Note that the testing here can be accomplished in conjunction with the testing of the security mechanisms directly. For example, testing performed to ensure that the administrative guidance provided is correct verifies that AGD_OPE.1 is satisfied and should address the invocation of the administrative actions that are needed to verify the audit records are generated as expected.

FAU_GEN.1.2

The TSF shall record within each audit record at least the following information:
a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and
b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, information specified in column three of **Table 2 Auditable Events**.

Assurance Activity:

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

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 [***no other key sizes***] 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 PUB 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-224, SHA-256, SHA-384, SHA-512***] and message digest sizes [***160, 224, 256, 384, 512***] bits that meet the following: FIPS Pub 180-3, 'Secure Hash Standard.'

Assurance Activity:

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

5.2.2.6 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, SHA-224, SHA-256, SHA-384, SHA-512***], key size [**equal to the block size**], and message digest sizes [***160, 224, 256, 384, 512***] 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 Explicit: HTTPS (FCS_HTTPS_EXT.1)

FCS_HTTPS_EXT.1.1

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

FCS_HTTPS_EXT.1.2

The TSF shall implement HTTPS using TLS as specified in FCS_TLS_EXT.1.

Assurance Activity:

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

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

FCS_RBG_EXT.1.1

The TSF shall perform all random bit generation (RBG) services in accordance with [***FIPS Pub 140-2 Annex C: X9.31 Appendix 2.4 using AES***] 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.

5.2.2.9 Explicit: SSH (FCS_SSH_EXT.1)

FCS_SSH_EXT.1.1

The TSF shall implement the SSH protocol that complies with RFCs 4251, 4252, 4253, 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.

5.2.2.10 Explicit: TLS (FCS_TLS_EXT.1)

FCS_TLS_EXT.1.1

The TSF shall implement one or more of the following protocols [***TLS 1.0 (RFC 2246),***] 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: "“!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(”, “)”,[**“‘”, “+”, “,”, “-”, “.”, “/”, “:”, “;”, “<”, “=”, “>”, “[”, “\”, “]”, “_”, “`”, “{”, “}”, and “~”**];
2. Minimum password length shall settable by the Security Administrator, and support passwords of 15 characters or greater;

Assurance Activity:

The evaluator shall examine the operational guidance to determine that it provides guidance to security administrators on the composition of strong passwords, and that it provides instructions on setting the minimum password length. The evaluator shall also perform the following tests. Note that one or more of these tests can be performed with a single test case.

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

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

FIA_UAU_EXT.2.1

The TSF shall provide a local password-based authentication mechanism, [***[and public/private key authentication]***] to perform 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.

5.2.4.4 User Identification and Authentication (FIA_UIA_EXT.1)

FIA_UIA_EXT.1.1

The TSF shall allow responses to 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;
- [***respond to ICMP echo requests***].

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.

5.2.5 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 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; and TLS/HTTPS; then all three 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 Basic Internal TSF Data Transfer Protection (FPT_ITT.1)

FPT_ITT.1.1

Refinement: The TSF shall protect TSF data from disclosure and detect its modification when it is transmitted between separate parts of the TOE through the use [**TLS/HTTPS**].

Assurance Activity:

The evaluator shall examine the TSS to determine that the methods and protocols used to protect distributed TOE components are described. 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 communication paths 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) communications 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: The evaluator shall ensure, for each method of communication, the channel data is not sent in plaintext.

Test 3: The evaluator shall ensure, for each method of communication, modification of the channel data is detected by the TOE.

Further assurance activities are associated with the specific protocols.

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

FPT_SKP_EXT.1.1

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

Assurance Activity:

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

5.2.6.3 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.

5.2.6.4 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.

5.2.6.5 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.

5.2.6.6 Extended: Trusted Update (FPT_TUD_EXT.1)

FPT_TUD_EXT.1.1

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

FPT_TUD_EXT.1.2

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

FPT_TUD_EXT.1.3

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

Component Assurance Activity:

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

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

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

5.2.7 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 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

Refinement: 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 or TLS/HTTPS***] **to** provide a 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

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

FTP_TRP.1.3

Refinement: 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 NDPP components as specified in Part 3 of the Common Criteria. Note that the SARs have effectively been refined with the assurance activities explicitly defined in association with both the SFRs and SARs.

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

Table 3 NDPP 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.

Component Assurance Activity:

There are no specific assurance activities associated with these SARs. The functional specification documentation is provided to support the evaluation activities described in Section 4.2 (of the NDPP), and other activities described for AGD, ATE, and AVA SARs. The requirements on the content of the functional specification information is implicitly assessed by virtue of the other

assurance activities being performed; if the evaluator is unable to perform an activity because the there is insufficient interface information, then an adequate functional specification has not been provided.

5.3.2 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 fulfill the security objectives for the operational environment as described in the ST.

AGD_OPE.1.7c

The operational user guidance shall be clear and reasonable.

AGD_OPE.1.1e

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

Component Assurance Activity:

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

Component Assurance Activity:

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

5.3.3 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.

Component Assurance Activity:

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

5.3.3.2 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.

Component Assurance Activity:

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

5.3.4 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.

Component Assurance Activity:

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

The test plan identifies the platforms to be tested, and for those platforms not included in the test plan but included in the ST, the test plan provides a justification for not testing the platforms. This justification must address the differences between the tested platforms and the untested platforms, and make an argument that the differences do not affect the testing to be performed. It is not

sufficient to merely assert that the differences have no affect; rationale must be provided. If all platforms claimed in the ST are tested, then no rationale is necessary.

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

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

5.3.5 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.

Component Assurance Activity:

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

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 audit records using the features available in its FreeBSD operating system kernel core as well as application-level audit events at the command level. As such, audit records are stored locally and are available to be exported using syslog-ng, which includes and implementation of TLSv1.0 for communication security. All the exported logs use a remote syslog-ng server. Logs are exported to the syslog-ng server instantly one at a time as they happen and regardless of whether a local copy can be saved. The TLSv1.0 network connection to the syslog-ng server is independent of local filesystem space availability for local copies of audit events. If the TLS connection drops, the audit events are buffered up to 100,000 events in memory and these buffered events will be sent once the connection to the syslog server is up again. The reconnection is automatic and tried every 10 seconds. If the network connection to the syslog-ng server was down for enough time (hours or days) so that the 100,000 event buffer limit is exceeded, then the local filesystem copy of the logs is the fallback copy. It can be copied off the box using SSH/SCP.

Locally stored audit records are bounded by available disk space which is variable, depending on consumption by the audit and other functions of the TOE. However, under normal conditions, there is more than enough space to store generated audit data for seven days, after which audit records older than seven days are deleted by the TOE. When the available disk space becomes around 80-90% full, the TOE reduces some of its operations so as to mitigate the possibility of complete disk space exhaustion. Some types of audit logs (those generated via GUI activity) can be configured to either wrap or not wrap the audit events. To wrap a log file is synonymous with overwriting old events with new events. If the TOE is not configured to wrap audit events then the audit log file will theoretically grow indefinitely. For this reason, it is recommended that wrapping be configured; and that an external syslog server(s) be configured. The TOE offers Authorized Administrators the ability to delete the audit logs as necessary. If the available space does become exhausted, the TOE operating system is designed to continue operating by default, but the TOE might not be able to record audit events locally until more space is made available for recording audit events though they will still be sent to the syslog server (if configured).

Among the auditable events are security relevant events such as start-up and shutdown; success and failure login of the user, regardless of the authentication mechanism; changing a user's password; adding and deleting user accounts; and any privileged (e.g., administrator) commands. 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 log files (internal to the TOE appliance) that is protected so that only an authorized administrator can read (for which tools accessible via the Console and Web Management Interface are provided) or otherwise access them. The protection results from the fact that the logs can be accessed only after an administrator logs in (see section 6.4 below).

The TOE includes a hardware clock that is used to provide reliable time information for the audit records it generates. The TOE can be configured to periodically synchronize its clock with a time server (using NTP), but the TOE can only ensure its own reliability and not that of an external time mechanism.

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 that caused the event.
- FAU_STG_EXT.1: The TOE can be configured to export audit records to an external audit server using syslog-ng based on TLSv1.0.

6.2 Cryptographic support

The TOE includes OpenSSL FIPS Object Module providing supporting cryptographic functions. The following functions have been CAVP tested in accordance with the identified standards.

Functions	Standards	Certificates
Asymmetric key generation		
• Domain parameter generation and key establishment	NIST Special Publication 800-56B	Cert. http://csrc.nist.gov/groups/STM/cavp/documents/dss/rsaval.html #1336
Encryption/Decryption		
• AES CBC (128, and 256 bits)	FIPS Pub 197 NIST SP 800-38A	Cert. http://csrc.nist.gov/groups/STM/cavp/documents/aes/aesval.html #2612
Cryptographic signature services		
• RSA Digital Signature Algorithm (rDSA) (modulus 2048)	FIPS Pub 186-2	Cert. http://csrc.nist.gov/groups/STM/cavp/documents/dss/rsaval.html #1336
Cryptographic hashing		
• SHA-1, SHA-224, SHA-256, SHA-384, and SHA-512 (digest sizes 160, 256, 384, and 512 bits)	FIPS Pub 180-3	Cert. http://csrc.nist.gov/groups/STM/cavp/documents/shs/shaval.htm #2195
Keyed-hash message authentication		
• HMAC-SHA-1, HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 (digest size 160, 256, 384, and 512 bits)	FIPS Pub 198-1 FIPS Pub 180-3	Cert. http://csrc.nist.gov/groups/STM/cavp/documents/mac/hmacval.html #1619
Random bit generation		
• RGB with one independent software based noise source with a minimum of 256 bits of non-determinism	FIPS Pub 140-2 Annex C: X9.31 Appendix 2.4 using AES	Cert. http://csrc.nist.gov/groups/STM/cavp/documents/rng/rngval.html #1237

Table 4 Cryptographic Functions

These functions above have been tested under the CAVP program.

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 that publication along with an indication of how the TOE conforms to those conditions.

NIST SP800-56B Section Reference	"should", "should not", or "shall not"	Implemented?	Rationale for deviation
5.6	should	yes	Not applicable
5.8	shall not	no	Not applicable
5.9	shall not (first occurrence)	no	Not applicable
5.9	shall not (second occurrence)	no	Not applicable
6.1	should not	no	Not applicable
6.1	should (first occurrence)	yes	Not applicable
6.1	should (second occurrence)	yes	Not applicable
6.1	should (third occurrence)	yes	Not applicable
6.1	should (fourth occurrence)	yes	Not applicable
6.1	shall not (first occurrence)	no	Not applicable
6.1	shall not (second occurrence)	no	Not applicable
6.2.3	should	yes	Not applicable
6.6	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 FIPS Pub 140-2 Annex C: X9.31 Appendix 2.4 using AES. The entropy source is a minimum of 256-bits based on data from the FreeBSD /dev/random function. The FreeBSD implementation of /dev/random is based on a 256-bit variant of the Yarrow algorithm, depending on potential attackers not knowing the internal state (i.e., of the appliance) and being reseeded frequently using network and disk activities in order improve entropy, to produce a cryptographically secure pseudorandom stream.

Additionally, the TOE is designed to zeroize (i.e., overwrite with zeros) secret and private keys when they are no longer required by the TOE. This includes all TLS and SSH session encryption and integrity keys) as well as RSA

host and user private keys and Apache SSL host certificate). This key-zeroization is provided by the OpenSSL FIPS Module with the following secure features as per the OpenSSL SecurityPolicy-1.2: *Keys residing in internally* (RAM) *allocated data structures in the Module can only be accessed using the Module defined API. The operating system protects memory and process space from unauthorized access. Zeroization of sensitive data is performed automatically by API function calls for intermediate data items, and on demand by the calling process using Module provided API function calls provided for that purpose. Only the process that creates or imports keys can use or export them. No persistent storage of key data is performed by the Module.* TLS (used by Apache web server) and SSH (used by SSH daemon) exchange keys upon connection setup and the keys are zeroized when the connection is dropped. The TOE also stored a number of configurable public keys that are not subject to zeroizatiom, but rather are overwritten with new values when updated.

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

The TOE supports TLSv1.0 with AES (CBC) 128 or 256 bit ciphers, in conjunction with SHA-1, and RSA. The following cipher suites are implemented by the TOE: TLS_RSA_WITH_AES_128_CBC_SHA, TLS_RSA_WITH_AES_256_CBC_SHA, TLS_DHE_RSA_WITH_AES_128_CBC_SHA, and TLS_DHE_RSA_WITH_AES_256_CBC_SHA. Note that server authentication is the default, but the administrator can configure certificates in the console application on the server to authenticate users just as users can install certificates in their browser to authenticate the server. Hence, mutual authentication is supported. The TOE is shipped with a default self-signed certificate and administrators can install their own certificates subsequently; including SSL certificates, web client user certificates, and trusted certificate authority certificates.

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). The TOE implementation of SSHv2 is OpenSSH 5.8p2-11. While other ciphers and hashes are implemented in the product, they are disabled in the evaluated configuration.

The SSHv2 authentication timeout period is 120 seconds allowing clients to retry only 3 times; both public-key and password based authentication can be configured; and packets are limited to 256K bytes. Whenever the timeout period or authentication retry limit is reached, the TOE closes the applicable TCP connection and releases the SSH session resources. As SSH packets are being received, the TOE uses a buffer to build all packet information. Once complete, the packet is checked to ensure it can be appropriately decrypted. However, if it is not complete when the buffer becomes full (256K bytes) the packet will be dropped.

There are a number of optional SSH capabilities, mostly not security related, as identified throughout the applicable RFCs. While DSS is supported in addition to RSA, it has not been evaluated and should not be used since the NDPP requires the use of 2048 bits (as provided by RSA). Multiple authentication of users is not supported, nor is host-based authentication. Should a service or user name change or be unavailable during SSH authentication, the connection will be dropped. The TOE always requires user authentication, so a method of 'none' is not accepted and will not be returned as a supported value. Furthermore, the TOE will ignore data sent after the allowed channel window, ignore languages in the algorithm negotiation, and provides no support for configuring the compatibility flag.

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_HTTPS_EXT.1: The TOE supports TLS/HTTPS web-based secure administrator sessions.

- 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/HTTPS web-based secure administrator sessions.

6.3 User data protection

The TOE is designed to ensure its own internal integrity as well as to protect user data from potential, unintended reuse by clearing resources (e.g., memory) as they are allocated to create objects used in the implementation of the TOE operations.

Given the nature of the TOE, the primary user is administrators and the resources they access. The network scanning capabilities result in the collection of carefully managed data. Administrators use that data in accordance with available functions and when the associated storage resource is freed, any previous data is overwritten prior to becoming accessible with new data subsequently.

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 supports the definition of users within an embedded user database. Users are defined with a username, password, and role (i.e., privileges). By default users use certificates to authenticate the server, but users can also optionally be configured with certificates that can be used to authenticate the user when connecting via the Web interface. In order to log into the console (including the local login for the laptop TOE configuration) or establish an SSH or TLS/HTTP session in order to exercise available functions, the user must provide a valid username and provide either the correct corresponding password or certificate. All GUI users must be configured to all use either passwords or certificates. In other words if GUI PKI is turned on then all GUI users authenticate using certificates. If the GUI certificate fails there is no fallback and access is denied. SSH users can be configured individually to use either passwords; certificate; or both. If a SSH user is configured to authenticate using either method then the TOE will first attempt authentication using certificate. If certificate authentication fails then the TOE will request a valid password from the user. If the user is not defined or the authentication data is incorrect, the user will be denied access to the TOE. In any case, the TOE is designed to not echo passwords back to the user while they are being entered.

Note that the TOE is a network scanning device and as such will perform that function based on its prior configuration without the need for an applicable user to be continuously logged in, though the user would need to login in order to see the result.

The TOE implements a number of constraints when setting or changing passwords, limiting the passwords to specific characters (per FIA_PMG_EXT.1) and also requiring that passwords have a minimum length as previously configured by an administrator. These constraints require that the administrator enable password controls after which minimum and maximum password lengths can be defined (the defaults are 8 characters minimum and 32 character maximum). Note that SSH logins are always subject to password constraints and while the default minimum is 8 characters, it can be configured by an administrator to be higher (e.g., 15 characters).

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 as described above.

- FIA_UAU.7: The TOE does not echo passwords as they are entered; rather '*' characters are echoed when entering passwords.
- FIA_UAU_EXT.2: The TOE can be configured to utilize local user accounts with passwords and optionally PKI certificates for the purpose of authentication.
- FIA_UIA_EXT.1: The TOE doesn't offer any services or access to its functions, except for displaying login banners and responding to ICMP echo requests, without requiring a user to be identified and authenticated.

6.5 Security management

The TOE supports administrative privileges and hence an administrator role. Only administrators can define and assign privileges to users. In general, the TOE restricts all security management functions (including product updates and configuration of cryptographic parameters) to users with administrative privileges. While other users can be defined (i.e., without privileges or the administrative role), they cannot perform administrator functions since the TOE is designed to require specific privileges for each administrator operation.

Note that in the context of this Security Target the administrator (with administrative privileges) corresponds to the "Authorized Administrator" and "Security Administrator" as referenced in the requirements.

Security management functions can be performed via a directly connected console or remotely using SSH or TLS/HTTPS sessions once the applicable user has been identified and authenticated. In the case of a laptop, its built-in keyboard, mouse and display functions serve as the directly connected console.

When multiple, cooperating components are deployed, they are individually managed via their respective interfaces and configured to interoperate as determined by the administrator for each component.

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 administrative privilege.
- FMT_SMF.1: The TOE includes the functions necessary to enable/disable available network services, to manage the cryptomodule and associated functions, and to manage and verify updates of the TOE software and firmware.
- FMT_SMR.2: The TOE includes the authorized administrator role to restrict security management within the TOE.

6.6 Protection of the TSF

The TOE is an appliance and as such is designed to work independent of other components. Secure communication with third-party peers as addressed in section 6.8, Trusted path/channels, and secure communication among multiple instances of the TOE is protected using TLS.

While the administrative interface is function rich, the TOE is designed specifically to not provide access to locally stored passwords, including passwords used to authenticate peer TOE appliances (which happen to be protected using MD-5 hashing or a proprietary base64 encryption algorithm while stored internally in access-protected files nonetheless) and also, while cryptographic keys can be entered and are stored internally in access-protected files, the TOE does not disclose any private keys stored in the TOE. Note that while there is a symmetric AES key for decrypting updates stored internally in an access-protected file, symmetric keys are generally created and stored in memory while applicable sessions (SSH, TLS) are active and are not accessible since no TOE interface is provided to access the memory of running processes.

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 clock is used for audit record time stamps,

measuring system inactivity prior to termination, and also for cryptographic operations where time stamps are applicable.

The TOE includes a number of built in diagnostic tests that are run during start-up to determine whether the TOE is operating properly. The TOE will stop, with errors displayed, when an error is encountered. The built-in self tests include basic read-write memory (involving writing non-zero values to memory and ensuring they are read back correctly), disk read, software checksum tests, and device detection tests. Once the basic tests are completed and the TOE components begin to start, the central TOE GUI component is designed to query each other component to ensure it is started and able to respond to help identify any failed components. Each component independently checks to ensure the applicable (i.e., where it and its data files are stored) disks are responding and that the executable, library, and configuration files are present and accessible. If a disk crash occurs, periodic fsck test fails, or a component fails to respond or gain access to its files, the TOE will report 'failed to start' rather than 'started'. The power-on self-tests comply with the FIPS 140-2 requirements for self testing. Once the TOE is operating, the GUI component periodically polls the other TOE components (including distributed TOE resources) and displays green, yellow, red, and blue indicators of operation where green represents no problems, blue represents a paused state, and yellow and red indicate that some or all components are failing and actions should be taking. Together, these tests serve to determine and track the overall health and correct operation of the TOE.

The TOE supports both patching and complete software upgrades. Patches are initiated by an administrator who can import that patch from a URL via the GUI. When the TOE installs the patch, it will check the digital signature of the patch to ensure that the patch has originated from Lumeta and has not been modified. Similarly, the administrator can obtain an upgrade image (iso file) on a CD/DVD and initiate an update via the Console. The TOE will check the digital signature of the patch to ensure that the patch has originated from Lumeta and has not been modified prior to performing the upgrade. In each case, the digital signature is verified using the public key in the certificate configured in the TOE. If the signature cannot be verified, the update fails and cannot be installed. Note that while the TOE comes preinstalled with an applicable Lumeta public certificate.

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

- FPT_ITT.1: The communication between associated appliances is protected using TLS.
- 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 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 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 Lumeta.

6.7 TOE access

The TOE can be configured to display administrator-configured advisory banners that will appear under a variety of circumstances. A login banner can be configured to display welcome information displayed in conjunction with login prompts. A message of the day can also be configured to be displayed after authentication is completed. A system access notification message can be configured to present legal and other advisories prior to a user logging in and this banner waits, requiring the user to confirm whether they want to continue with the authentication process. In each case, the banners will be displayed when accessing the TOE via the Console (including both serial connections and the directly attached laptop human interface devices monitor, keyboard, and mouse) or GUI interfaces.

The TOE can be configured by an administrator to set a session timeout value (any integer value in minutes and also optionally in seconds, with 0 disabling the timeout) for GUI sessions. The default timeout for GUI sessions is 15 minutes. Additionally, the console and SSH shell sessions timeout after a default of 15 minutes of inactivity and is also configurable. 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. Of course, a user can otherwise terminate their interactive session any time they want.

The user will be required to login in after any session has been terminated due to inactivity or after voluntary termination.

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.

6.8 Trusted path/channels

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

In order to establish trust among communicating components, applicable certificates can be obtained from a configured Certificate Authority (establishing common trust) or alternate can be generated and self-signed and administratively installed where needed.

Remote access to audit records is protected using TLS implemented as part of syslog-ng.

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

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

7. Protection Profile Claims

The ST conforms to the *Security Requirements for Network Devices*, version 1.99, 8 June2012 (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

Since all assumptions, threats, organizational security policies, and security objectives have been drawn from the NDPP, the NDPP should be consulted for the correspondence mappings and rationale.

8.2 Security Requirements Rationale

Since all security objectives and requirements have been drawn from the NDPP, the NDPP should be consulted for the correspondence mappings and rationale.

8.3 Security Assurance Requirements Rationale

The Security Assurance Requirements (SARs) 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_HTTPS_EXT.1	FCS_TLS_EXT.1	FCS_TLS_EXT.1
FCS_RBG_EXT.1	none	none
FCS_SSH_EXT.1	FCS_COP.1	FCS_COP.1(*)
FCS_TLS_EXT.1	FCS_COP.1	FCS_COP.1(*)
FDP_RIP.2	none	none

ST Requirement	CC Dependencies	ST Dependencies
FIA_PMG_EXT.1	none	none
FIA_UAU.7	FIA_UAU.1	FIA_UIA_EXT.1
FIA_UAU_EXT.2	none	none
FIA_UIA_EXT.1	none	none
FMT_MTD.1	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.1 and FMT_SMF.1
FMT_SMF.1	none	none
FMT_SMR.1	FIA_UID.1	FIA_UIA_EXT.1
FPT_ITT.1	none	none
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 6 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 7 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**	**Resource utilisation**	**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_HTTPS_EXT.1		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_UAU_EXT.2				X					
FIA_UIA_EXT.1				X					
FMT_MTD.1					X				
FMT_SMF.1					X				
FMT_SMR.1					X				
FPT_ITT.1						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			
FRU_RSA.1							X		
FTA_SSL.3								X	
FTA_SSL.4									
FTA_SSL_EXT.1								X	
FTA_TAB.1								X	
FTP_ITC.1									X
FTP_TRP.1									X

Table 7 Security Functions vs. Requirements Mapping