Title:	Security Target (ST)
Version:	1.1
Date:	2014-02-28
Subject:	ST for Cisco Catalyst 6500-E Series Switches
Author:	Cisco Systems, Inc

©2014, Cisco Systems, Inc. All rights reserved. This document can be reproduced in full without any modifications.

Table of contents

Figures

Tables

1 ST introduction

The structure and content of this ST comply with the requirements specified in the Common Criteria (CC), Part 1, Annex A, and Part 3, Chapter 4.

The Security Target contains the following sections:

- Security Target Introduction [Section 1]
- Conformance Claims [Section 2]
- Security Problem Definition [Section 3]
- Security Objectives [Section 4]
- Extended Components Definition [Section 5]
- IT Security Requirements [Section 6]
- TOE Summary Specification [Section 7]
- Appendix [Section 8]

This Security Target (ST) defines a set of assumptions about the aspects of the environment, a list of threats that the product intends to counter, a set of security objectives, a set of security requirements, and the IT security functions provided by the TOE which meet the set of requirements.

1.1 ST reference

The ST reference provides identification material for the ST.

Title:	Security Target (ST) for Cisco Catalyst 6500-E Series Switches
Version:	1.1
Date:	2014-02-28
EDCS	1252106
Authors:	Cisco Systems, Inc
Registration:	
Certification ID:	BSI-DSZ-CC-0937
CC Version:	Version 3.1 Revision 3
Keywords:	Audit, Authentication, Encryption, , Protection, Switch, Traffic

Table 1: ST reference

1.2 TOE reference

The TOE reference provides identification material for the TOE that the ST refers to.

Title:	Cisco Catalyst 6500-E Series Switches
Version:	Hardware Models - Cisco Catalyst 6500-E Series Switches 6503-E, 6504-E, 6506-E, 6509-E, and 6513-E with Supervisor Engine 2T (Excluding Sup720) Software Version – IOS 15.1(1)SY1
Developer:	Cisco Systems, Inc
Registration:	-
Certification ID:	BSI-DSZ-CC-0937
CC Version:	3.1 Revision 3
Keywords:	Audit, Authentication, Encryption, Protection, Switch, Traffic

Table 2: TOE reference

1.3 TOE overview

The TOE is the Cisco Catalyst 6500-E Series Switches 6503-E, 6504-E, 6506-E, 6509-E, and 6513-E with Supervisor Engine 2T (Excluding Sup720) running IOS 15.1(1)SY1 (herein after referred to as the 6500-E Series, the switch, or the TOE). The TOE is a purpose-built, switching and routing platform with
110 OSI Layer2 and Layer3 traffic filtering capabilities.

Cisco IOS is a Cisco-developed highly configurable proprietary operating system that provides for efficient and effective routing and switching. Although IOS performs many networking functions, this TOE only addresses the functions that provide for the security of the TOE itself as described in Section 1.4.4 TOE logical scope below.

1.3.1 Usage and major security features of the TOE

The TOE consists of any one of a number of hardware configurations, each running the same version of IOS software. The 6500-E Series chassis provides power, cooling, and backplane for the Supervisor Engine, line cards, and service modules (SM)[1]. The Supervisor Engines run the IOS software. The evaluated configurations consist of the following (e.g. at least one of the listed chassis, at least one
120 supervisor cards running IOS 15.1(1)SY1 software and at least one line cards):

TOE	• One or more WS-C6503-E, WS-C6504-E, WS-C6506-E, WS-C6509-E, or WS-C6513-E Switch Chassis (Two chassis can be configured together to support HA with VSS.)
	• One or more Supervisor 2T (Sup2T) Cards (VS-S2T-10G or VS-S2T-10G-XL) per chassis (Two Sup cards in one chassis provide Supervisor failover within the chassis.)
	• Each Sup2T running IOS 15.1(1)SY1 (FIPS validated)
	• With one or more of the following Line Cards installed to one or more chassis:
	  • WS-X6908-10G-2TXL / WS-X6908-10G-2T
	  • WS-X6848-SFP-2TXL / WS-X6848-SFP-2T
	  • WS-X6824-SFP-2TXL / WS-X6824-SFP-2T
	  • WS-X6848-TX-2TXL / WS-X6848-TX-2T
	  • WS-X6816-10G-2TXL / WS-X6816-10G-2T
	  • WS-X6816-10T-2TXL / WS-X6816-10T-2T
	  • WS-x6904 Estelle-4x40 GE / 16x10 GE (Lite or XL)

Table 3: Evaluated Configuration

The TOE can optionally connect to an NTP server on its internal network for time services. If an NTP server is used, it must only be accessible via the internal network (an internal network isolated from user traffic and intended for use by TOE administrators only).

If the TOE is to be remotely administered, management must be through an IPSec tunnel.

The TOE will transmit syslog message to a remote syslog server through an IPSec tunnel.The TOE can also be configured to use a remote AAA server (RADIUS or TACACS+) for centralized authentication, and can also connect to those servers through an IPSec tunnel.

130 1.3.2 TOE type

The TOE is a switching and routing platform used to construct IP networks by interconnecting multiple smaller networks or network segments. As a Layer2 switch, it performs analysis of incoming frames, makes forwarding decisions based on information contained in the frames, and forwards the frames

[1] No specific service modules, such as the Firewall Blade, Wireless Service and Network Analysis being claimed in the evaluated configuration as they require additional license

toward the destination. As a Layer3 switch/router, it supports routing of traffic based on tables identifying available routes, conditions, distance, and costs to determine the best route for a given packet. Routing protocols used by the TOE include BGPv4, EIGRP, RIPv2, OSPFv2, and HSRP. EIGRP supports routing updates with IPv6 or IPv4, as does BGPv4, while RIPv2 and OSPFv2 routing protocol support routing updates for IPv4 only. HSRP is not a routing protocol; it is used in order to achieve default gateway failover if the primary gateway becomes inaccessible.

1.3.3 TOE life cycle

The TOE life cycle consists of several phases and is shown in the following figure:

Phase 1:	**Design and production**
↓	
Phase 2:	**Delivery and integration**
↓	
Phase 3:	**Initialization procedure**
↓	
Phase 4:	**Administration and operational practice**
↓	
Phase 5:	**Decommissioning procedure**

Figure 1: TOE life cycle phases

The TOE as defined in this ST exists first and only during TOE life cycle phase 4, where the TOE is administered and operated by its intended users. TOE life cycle phase 4 is covered by CC assurance class guidance documents (AGD), especially the CC assurance family operational user guidance (AGD_OPE).

TOE life cycle phase 1 is covered by CC assurance class life cycle support (ALC).

TOE life cycle phases 2 and 3 are covered by ALC and AGD, especially the CC assurance family preparative procedures (AGD_PRE).

TOE life cycle phase 5 is covered by AGD.

1.3.4 Required non-TOE hardware/software/firmware

The TOE supports (in some cases optionally) the following hardware, software, and firmware in its environment:

Component	Required	Usage/Purpose Description for TOE performance
Management Workstation	Yes	This includes any IT Environment Management workstation that is used by the TOE administrator to support TOE administration through protected channels (e.g. IPSec).

Component	Required	Usage/Purpose Description for TOE performance
NTP Server	No	The TOE supports communications with an NTP server to receive clock updates. Any server that supports NTPv1 (RFC 1059), NTPv2 (RFC 1119), or NTP v3 (RFC 1305) may be used.
Syslog server	Yes	The syslog audit server is used for remote storage of audit records that have been generated by and transmitted from the TOE.
Authentication Server	Yes	The authentication server (RADIUS RFC 2865, 2866, 2869 and RFC 3162 (IPv6) and TACACS+ RFC 1492)) is used to provide centralized authentication and related auditing for one or more distributed instances of the TOE.

Table 4: IT Environment Components

1.4 TOE description

The TOE description explains the TOE in more detail than was provided in the TOE overview.

1.4.1 TOE architecture and security capabilities

The TOE architecture and security capabilities show the layout of the TOE and give a general understanding of the security capabilities of the TOE.

The TOE chassis can be deployed in a data center, or in a wiring closet, at the distribution and core layers, and at the WAN edge, providing the power and features required for end-to-end deployment for the enterprise campus, the ISP network, and metro and research computing networks.

Chassis Applications -

Cisco IOS Software and Cisco Catalyst Operating System Software are supported across all supervisor engines

Layer 2 and Layer 3 quality of service (QoS), facilitating tiered Ethernet service offerings through rate limiting and traffic shaping

The TOE provides 3-, 4-, 6-, 9-, and 13-slot chassis models with slots arranged horizontally, and a 9-slot model with slots arranged vertically, with front-to-back airflow. Typical applications for the chassis include:

- 3-and 4-Slot Chassis: Low-density, wiring-closet chassis sharing interface modules and supervisor engines with larger chassis for common sparing; low-density, high-performance specialized services module chassis for network security and management; and low-density, high-end chassis providing connectivity to the WAN edge
- 6- and 9-Slot Chassis: Traditional chassis for the wiring closet, distribution and core layers, data center, and WAN edge. The Cisco Catalyst 6506-E and Catalyst 6509-E support more than 4000 watts (W) power and higher per slot.
- 13-Slot Chassis: Highest-capacity chassis for Ethernet connectivity, with slots to spare for services modules, providing network security and management.

Chassis Configuration -

Supports up to 576 10/100/1000 gigabit-over-copper ports or 1152 10/100 Ethernet ports

Features the industry's first 96-port 10/100 RJ-45 module, with optional, field-upgradable support for 802.3af PoE

Provides up to 192 Gigabit Ethernet ports

Redundant Supervisor Engines (stateful failover)

All Cisco Catalyst 6500-E Series chassis are NEBS Level-3 compliant and use common power supplies. The 6- and 9-slot chassis require a 1000W or 1300W power supply and the 13-slot

chassis requires a 2500W or 4000W power supply. The 3-slot chassis requires a 950W power supply.

Power -

All Cisco Catalyst 6500-E Series chassis holds up to two load-sharing, fault-tolerant, hot-swappable AC or DC power supplies. Only one supply is required to operate a fully loaded chassis. If a second supply is installed, it operates in a load-sharing capacity. The power supplies are hot-swappable-a failed power supply can be removed without powering off the system.

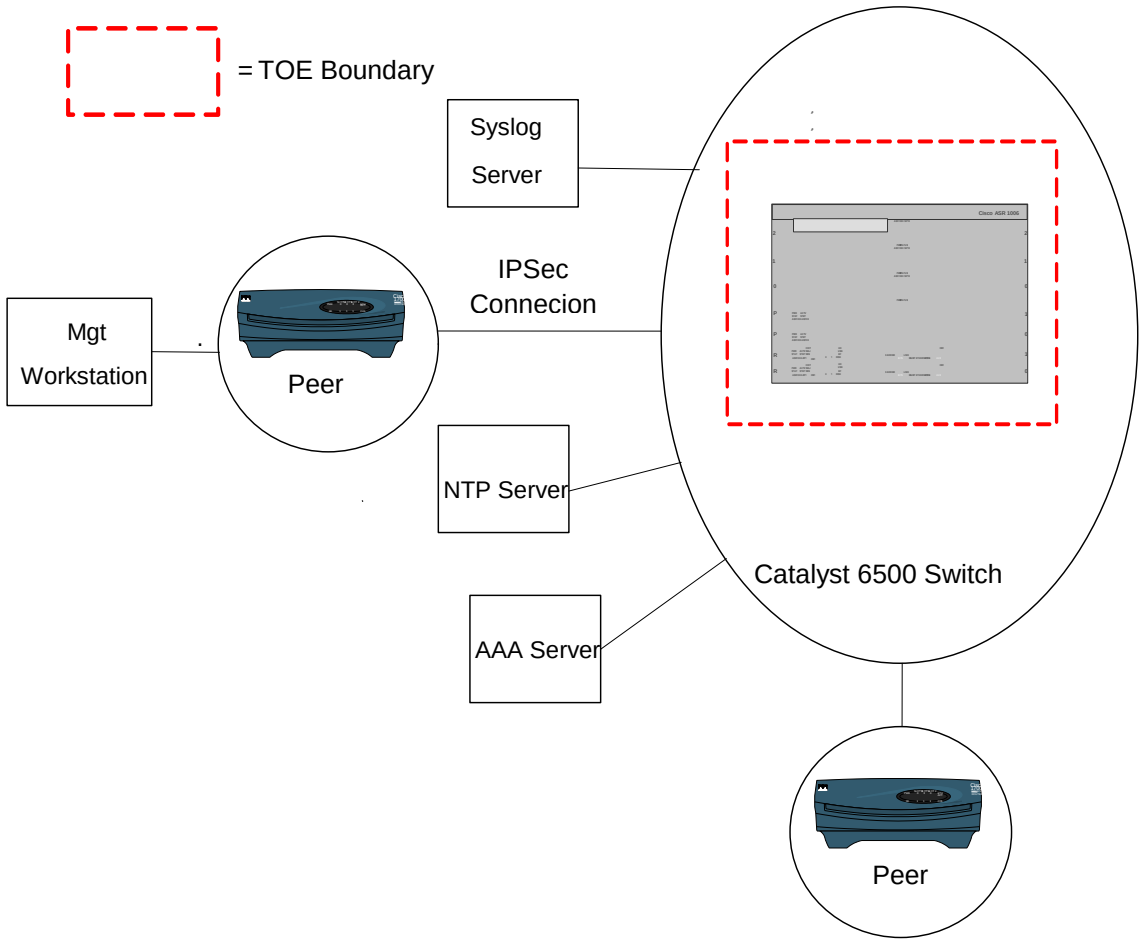
Cisco Catalyst 6500-E Series switch power supplies are available in the following power ratings:

- 950W AC input (Cisco Catalyst 6503-E chassis)
- 1400W AC input (Cisco Catalyst 6503-E chassis)
- 1000W AC input
- 1300W AC and DC input
- 2500W AC and DC input
- 2700W AC and DC input (Cisco Catalysist 6504-E chassis)
- 3000W AC input
    - 4000W AC input
    - 6000W AC input
    - 8700W AC input

Figure 2: Cisco Catalyst 6500-E Series Chassis

1.4.2 TOE environment and configuration

The TOE environment and configuration explain the wider application context into which the TOE will fit.



220 **Figure 3: TOE environment**

The physical boundary of the TOE is the switch hardware and software. The software of the TOE includes IOS and other supporting functionality (e.g., SSH Server). This physical boundary represents the Switch subsystem of the TOE. The Switch subsystem processes data packets and accepts a management interface connection to administer the switch. The management interface is either through a secure IPSec tunnel or via a local console connection.

The switches are hardware platforms in which all operations in the TOE scope are protected from interference and tampering by untrusted subjects. All administration and configuration operations are performed within the physical boundary of the TOE. Also, all TOE Security Policy (TSP) enforcement functions must be invoked and succeed prior to functions within the TOE scope of control (TSC)
230 proceeding.

The TOE includes a chassis, one or more supervisor engine cards running IOS 15.1(1)SY1 and one or more line cards. Each switch is a physical device with the following types of communication interfaces provided by the supervisor engine cards and/or the line cards:

- USB ports,
- Network port,
- Serial port, and

- Compact Flash Slot.

In addition to the communication interfaces above, the TOE includes a number of LEDs and power connectors. The LEDs are output elements only, and while the power connectors provide physical input they are not considered TOE interfaces.

1.4.2.1 USB Console Port

The USB Interface is a physical port on the Sup2T. The interface allows a console to be connected to the TOE as a USB device. Physical access to the port is protected by operational environment of the switch.

1.4.2.2 Network Ports

The physical network interfaces to the switch are Ethernet interfaces receiving and transmitting Internet Protocol datagrams as specified in RFC 0894 [**Ethernet**], RFC 0791 [**IPv4**], and RFC 2460 [**IPv6**]. Through these physical interfaces network traffic is transferred into and out of the TOE. The physical network interface (ports) can be located on the supervisor card and/or the line cards.

The network interface is the physical Ethernet interface to the TOE from the internal and external networks. Within the scope of the evaluation, these interfaces are used for the following purposes:

- For network traffic entering and leaving the TOE. This could be 'through traffic' for example a telnet packet from a user destined from an internal network to an external network, or 'to the box traffic' for example an external ping to the TOE's IP address.
- To allow a remote Administrator to access the TOE's CLI over the network using secure IPSec tunnel.
- To allow the TOE to establish IPSec tunnels with VPN peers and transmit syslog messages through those tunnels.
- To allow the TOE to use NTP to synchronize its clock with a time server.
- To allow the TOE access a AAA server to authenticate TOE administrators.

1.4.2.3 Serial Port

From a directly connected terminal an Administrator can authenticate to the TOE and issue commands to the TOEs CLI. This interface can be configured to display syslog messages to the console.

The primary serial interface into the TOE uses RS-232 signaling over an RJ45 interface. The serial port is located on the supervisor card.

The supervisor card also has a Management RJ45 connector and LED for the Connectivity Management Processor (CMP). It is noted that the CMP is to only be used when directed by Cisco Support personnel.

The management of the TOE is performed via the CLI either through the serial port or through an IPSec tunnel.

1.4.2.4 Compact Flash Slot

The Supervisor Engine card in the Catalyst 6500-E series provides a slot to accept a compact flash drive. The TOE can accept 64MB, 128 MB, 256 MB, 512 MB compact flash drives. The storage provided by these drives is used by the TOE as ordinary long term storage of configuration files and IOS software images.

Because the TOE treats the compact flash storage as an internal storage medium, this physical interface is considered internal to the TOE and thus, NOT a TSFI.

1.4.3 TOE physical scope

The TOE physical scope provides a collection of all hardware, firmware, software and guidance parts that constitute the TOE:

- The TOE is a hardware and software solution that uses a combination of chassis, supervisor engine, and line cards: the Cisco Catalyst 6500-E Series Switches 6503-E, 6504-E, 6506-E, 6509-E, and 6513-E with Supervisor Engine 2T (Excluding Sup720) running IOS 15.1(1)SY1 on the Supervisor Engine.
- Installation and Configuration guidance for the Common Criteria Evaluated Cisco Catalyst 6500-E Series Switches with IOS 15.1(1)SY1
- Cisco IOS Security Command Reference
- Cisco IOS Security Configuration Guide

1.4.4 TOE logical scope

The TOE is comprised of several security features. Each of the security features identified above consists of several security functionalities, as identified below.

1. Security audit
2. Cryptographic support
3. User data protection
4. Identification and authentication
5. Security Management (Access Control)
6. Protection of the TSF
7. Resource Utilization
8. TOE access
9. Trusted Path/Channels

These features are described in more detail in the subsections below.

1.4.5 Security Audit

The TOE generates a comprehensive set of audit logs that identify specific TOE operations. For each event, the TOE records the date and time of each event, the type of event, the subject identity, and the outcome of the event. Auditable events include: failure on invoking cryptographic functionality; establishment, termination and failure of an IPsec SA; modifications to the group of users that are part of the authorized administrator roles; all use of the user identification mechanism; any use of the authentication mechanism; any change in the configuration of the TOE; detection of replay attacks, changes to time, initiation of TOE update, indication of completion of TSF self-test, maximum sessions being exceeded, termination of a remote session and attempts to unlock a termination session; initiation and termination of a trusted channel; any matching of packets to access control entries in ACLs when traversing the TOE; and any failure of a packet to match an access control list (ACL) rule allowing traversal of the TOE.

The TOE is configured to transmit its audit messages to an external syslog server. Communication with the syslog server is protected using IPSec.

The logs can be viewed on the TOE using the appropriate IOS commands. The records include the date/time the event occurred, the event/type of event, the user associated with the event, and additional information of the event and its success and/or failure. The TOE does not have an interface to modify audit records, though there is an interface available for the authorized administrator to clear audit data stored locally on the TOE.

1.4.6 Cryptographic Support

The TOE provides cryptography support for secure communications and protection of information when configured in FIPS mode of operation. The crypto module is FIPS 140-2 SL2 validated. The cryptographic services provided by the TOE include: symmetric encryption and decryption using AES; digital signature using RSA; cryptographic hashing using SHA1; keyed-hash message authentication using HMAC-SHA1, and IPSec for authentication and encryption services to prevent unauthorized viewing or modification of data as it travels over the external network. The TOE also implements IPSec for secure remote administration. In the evaluated configuration, the TOE must be operated in FIPS mode of operation per the FIPS Security Policy (certificate 2002).

1.4.7 User Data Protection

The TOE supports routing protocols including BGPv4, EIGRP, RIPv2, and OSPFv2 (HSRP does not update routing tables) to maintain routing tables, or routing tables can configured and maintained manually ('static routes'). Since routing tables are used to determine which egress ACL is applied to the outbound traffic, the authority to modify the routing tables is restricted to authenticated administrators, and authenticated neighbor routers. The only aspect of routing protocols that is security relevant in this TOE is the TOE's ability to authenticate neighbor routers using shared passwords. Other security features and configuration options of routing protocols are beyond the scope of this Security Target and are described in administrative guidance.

The TOE also ensures that packets transmitted from the TOE do not contain residual information from previous packets. New packets that do not contain sufficient information to fill the minimum size of the data portion of the packet use zeros for padding the remainder of the packet so that residual data from previous traffic is never transmitted from the TOE.

1.4.8 Identification & Authentication

The TOE performs authentication, using Cisco IOS platform authentication mechanisms, to authenticate access to user EXEC and privileged EXEC command modes. All users wanting to use TOE services are identified and authenticated prior to being allowed access to any of the services. Once a user attempts to access the management functionality of the TOE (via EXEC mode), the TOE prompts the user for a user name and password. Only after the administrative user presents the correct identification and authentication credentials will access to the TOE functionality be granted.

The TOE also supports use of a remote AAA server (RADIUS and TACACS+) as the enforcement point for identifying and authenticating users attempting to connect to the TOE's CLI. Note the remote authentication server is not included within the scope of the TOE evaluated configuration, it is considered to be provided by the operational environment.

The TOE can be configured to display an advisory banner when administrators log in and also to terminate administrator sessions after a configured period of inactivity.

The TOE also supports authentication of other routers using router authentication supported by BGPv4, EIGRP, RIPv2, OSPFv2, and HSRP. Each of these protocols supports authentication of packets through use of MD5 hashing using shared passwords, which each neighbor router uses to authenticate packets upon receipt. For additional security, it is recommended router protocol traffic also be isolated to separate VLANs.

1.4.9 Security Management

The TOE provides secure administrative services for management of general TOE configuration and the security functionality provided by the TOE. All TOE administration occurs either through a secure IPSec tunnel, or a local console connection (serial port). The TOE provides the ability to perform the following actions:

- allows authorized administrators to add new administrators,

- start-up and shutdown the device,
- create, modify, or delete configuration items,
- create, modify, or delete information flow policies,
- create, modify, or delete a routing table,
- modify and set session inactivity thresholds,
- modify and set the time and date,
- and create, delete, empty, and review the audit trail

All of these management functions are restricted to authorized administrators of the TOE.

The TOE switch platform maintains administrative privilege levels and supports non-administrative connections. Non-administrative connections are established with authenticated neighbor routers for the ability to transmit and receive routing table updates per the information flow rules. No other access or management functionality is associated with non-administrative connections. The administrative privilege levels include:

- Administrators are assigned to privilege levels 0 and 1. Privilege levels 0 and 1 are defined by default and are customizable. These levels have a very limited scope and access to CLI commands that include basic functions such as login, show running system information, turn on/off privileged commands, logout.
- Semi-privileged administrators equate to any privilege level that has a subset of the privileges assigned to level 15; levels 2-14. These levels are undefined by default and are customizable.
- Privileged administrators are equivalent to full administrative access to the CLI, which is the default access for IOS privilege level 15.

The term “authorized administrator” is used in this ST to refer to any user account that has been assigned to a privilege level that is permitted to perform the relevant action; therefore has the appropriate privileges to perform the requested functions.

1.4.10 Protection of the TSF

The TOE protects against interference and tampering by untrusted subjects by implementing identification, authentication and access controls to limit configuration to authorized administrators. Additionally Cisco IOS is not a general-purpose operating system and access to Cisco IOS memory space is restricted to only Cisco IOS functions.

The TOE provides secure transmission when TSF data is transmitted between separate parts of the TOE (HA failover connectivity between two Sup2T cards in the same chassis security traverses the chassis backplane, and between HA failover Sup2T cards in separate chassis uses VSS, secured with VSL). Use of separate VLANs is used to ensure routing protocol communications between the TOE and neighbor routers including routing table updates and neighbor router authentication will be logically isolated from traffic on other VLANs.

The TOE is also able to detect replay of information received via secure channels (IPSec). The detection applied to network packets that terminate at the TOE, such as trusted communications between the administrators and the TOE, or between an IT entity (e.g., authentication server) and the TOE. If replay is detected, the packets are discarded.

In addition, the TOE internally maintains the date and time. This date and time is used as the timestamp that is applied to audit records generated by the TOE. Administrators can update the TOE’s clock manually, or can configure the TOE to use NTP to synchronize the TOE’s clock with an external time source. Finally, the TOE performs testing to verify correct operation of the switch itself and that of the cryptographic module.

1.4.11 Resource Utilization

The TOE provides the capability of controlling and managing resources so that a denial of service will not occur. The resource allocations are configured to limit the number of concurrent administrator sessions.

1.4.12 TOE Access

The TOE can terminate inactive sessions after an authorized administrator configurable time-period. Once a session has been terminated the TOE requires the user to re-authenticate to establish a new session.

420 The TOE can also display a Security Administrator specified banner on the CLI management interface prior to allowing any administrative access to the TOE.

1.4.13 Trusted Path/Channels

The TOE establishes a trusted path between the appliance and the CLI and the syslog server using IPSec encrypted connection. The TOE can also establish trusted paths of peer-to-peer VPN tunnels.

1.5 Excluded Functionality

The Cisco IOS contains a collection of features that build on the core components of the system. Those features that are not within the scope of the evaluated configuration include:

Featues that must remain disabled in the evaluated configuration:

430
- HTTP or HTTPS Server - The IOS web server (using HTTPS or HTTP) cannot satisfy all the NDPP requirements for administrative interfaces and must remain disabled in the evaluated configuration. The CLI interface is used to manage the TOE. Not including this feature does not interfere with the management of TOE as defined in the Security Target or the operation of the TOE.
- IEEE 802.11 Wireless Standards requires additional hardware beyond what is included in the evaluated configuration.
- SNMP Server does not enforce the required user-specific authentication. This feature is disabled by default and must remain disabled in the evaluated configuration. Including this feature would not meet the security policies as defined in the Security Target. The exclusion of
440 this feature has no effect on the operation of the TOE.
- Telnet server sends authentication data in the clear. This feature is enabled by default and must be disabled in the evaluated configuration. Including this feature would not meet the security policies as defined in the Security Target. The exclusion of this feature has no effect on the operation of the TOE.
- VPN Remote Access requires additional licenses beyond what is included in the evaluated configuration. Administrative remote access is secured using IPSec.
- Smart Install is a feature to configure IOS Software and switch configuration without user intervention. The Smart Install uses dynamic IP address allocation to facilitate installation providing transparent network plug and play. This feature is not to be used as it could result in
450 settings/configurations that may interfere with the enforcement of the security policies as defined in the Security Target.
- CMP (Connectivity Management Processor) interface provides a backup network interface to the supervisor engine when the main Route Processor (RP) is unreachable. The CMP should only be used under the guidance of Cisco Support Personnel.
- TrustSec is only relevant to RADIUS KeyWrap, which is being represented with other cryptographic methods identified and described in this Security Target. This feature is disabled by default and should remain disabled in the evaluated configuration. Not including this feature does not interfere with the enforcement of the security policies as defined in the Security Target or the TOEs operation.

460

Apart from these exceptions, all types of network traffic through and to the TOE are within the scope of the evaluation.

2 Conformance claims

The conformance claims indicate the source of the collection of requirements that is met by the ST.

2.1 CC conformance claims

The ST is conformant to CC Version 3.1 Revision 3 [CC].

The ST is CC Part 2 extended. The following security functional requirements (SFRs) are not based upon functional components in CC Part 2 [CC-P2]; the SFRs are from the US Government, Security Requirements for Network Devices (pp_nd_v1.0), version 1.0, dated 10 December 2010 [NDPP]:

470 FAU_STG_EXT.1, FAU_STG_EXT.3, FCS_CKM_EXT.4, FCS_COMM_PROT_EXT.1, FCS_IPSEC_EXT.1, FCS_RBG_EXT.1, FIA_PMG_EXT.1, FIA_UIA_EXT.1, FIA_UAU_EXT.5, FPT_TUD_EXT.1, FPT_TST_EXT.1, and FTA_SSL_EXT.1.

The ST is CC Part 3 conformant. All security assurance requirements (SARs) are based only upon assurance components in CC Part 3 [CC-P3].

2.2 PP claims

This ST claims strict conformance to the following Common Criteria validated Protection Profiles (PP), US Government, Security Requirements for Network Devices (pp_nd_v1.0), version 1.0, dated 10 December 2010 (from here within referred to as NDPP). To support the strict conformance claim, as noted below in the PP conformance claim rationale, the ST includes all claims as indicated in NDPP and makes no
480 additional claims.

This Security Target has adapted the Security Problem Definition, Security Objectives, and Security Functional Requirements (SFRs) from the NDPP.

2.3 Package claims

The ST claims conformance to the NDPP and has adapted the Security Problem Definition, Security Objectives, and Security Functional Requirements (SFRs).

2.4 Conformance rationale

The conformance rationale demonstrates why the chosen conformance claims were deemed appropriate.

This Security Target claims conformance to the NDPP and as such TOE provides all of the functionality at a level of security commensurate with that identified in the NDPPv1.0.

490 In addition, the TOE type stated in this Security Target is consistent with the TOE type stated in the NDPPv1.0 to which the Secuirty Target is claimed to be conformant.

Furthermore, this Security Target also adapts the Security Problem Definitions (SDP) that includes the threats, Organizational Security Policy(s), Assumptions, and Objectives, as well as the Security Functional Requirements (SFR) and Security Assurance Requirements (SAR) exactly from the NDPP verbatim.

3 Security problem definition

The security problem definition (SPD) defines the security problem that is to be addressed.

3.1 Introduction

This section describes the security environment in which the TOE is intended to be used.

3.2 External entities

The following human or IT entities possibly interact with the TOE from outside the TOE boundary.

Admin	Human who administers and uses the TOE. Administration tasks include starting the TOE, operating the TOE, maintaining configuration data, inspection of security audit log files and shut down the TOE. In this Security Target there are several levels of administrators, all which are described in Section 7.5 and all considered an Admin.
Attacker	A threat agent trying to undermine the security policy of the TOE.

Table 5: External entities interacting with TOE

3.3 Assets

The owner of the TOE presumably places value upon the following primary and secondary entities as long as they are in the scope of the TOE.

3.3.1 Primary assets

The owner of the TOE presumably places value upon the following primary entities. All these primary assets represent user data in the sense of the CC.

Audit data	Primary asset, audit data The data which is provided by the TOE during security audit logging. Security properties to be maintained by the TOE: confidentiality, availability, integrity.

Table 6: Primary assets to be protected

3.3.2 Secondary assets

The owner of the TOE presumably places value upon the following secondary entities. All these secondary assets represent TSF and TSF data in the sense of the CC.

Auth data	Secondary asset, TSF data The data which is used by the TOE to identify and authenticate the external entities which interact with the TOE. Security properties to be maintained by the TOE: confidentiality, integrity, authenticity.
Crypto data	Secondary asset, TSF data The data which is used by the TOE for digital signature handling and encryption/decryption purposes. Security properties to be maintained by the TOE: confidentiality, integrity, authenticity.
Ctrl data	Secondary asset, TSF data The data which is used by the TOE for firmware updates, firmware registration, and firmware identity checking purposes.

	Security properties to be maintained by the TOE: availability, integrity.

Table 7: Secondary assets to be protected

3.4 Assumptions

Assumptions are made on the operational environment of the TOE in order to be able to provide security functionality.

3.4.1 Assumptions on physical aspects of the environment

The following assumptions apply to physical aspects of the environment.

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

Table 8: Assumptions on physical aspects of the environment

3.4.2 Assumptions on personnel aspects of the environment

The following assumptions apply to personnel aspects of the environment.

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

Table 9: Assumptions on personnel aspects of the environment

3.4.3 Assumptions on connectivity aspects of the environment

The following assumptions apply to connectivity aspects of the environment.

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.

Table 10: Assumptions on connectivity aspects of the environment

3.5 Threats

This chapter holds a collection of threats that are to be countered by the TOE, its operational environment, or a combination of the two. A threat consists of an adverse action performed by a threat agent on assets.

T.ADMIN_ERROR	An administrator may unintentionally install or configure the TOE incorrectly, resulting in ineffective security mechanisms. Threat agent: Administrator. Asset at risk: Auth, crypto, ctrl data. Adverse action 1: impact to security Adverse action 2: unintential actions
T.RESOURCE_EXHAUSTION	A process or user may deny access to TOE services by exhausting critical resources on the TOE. Threat agent: Attacker. Asset at risk: Auth, crypto, ctrl data. Adverse action 1: denial of service (DoS) Adverse action 2: resource consumption
T.TSF_FAILURE	Security mechanisms of the TOE may fail, leading to a compromise of the TSF.

	Threat agent: Attacker. Asset at risk: Auth, crypto, ctrl data. Adverse action 1: Failure of mechanisms (primitive/complex) Adverse action 2:
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. Threat agent: Attacker. Asset at risk: Auth, crypto, ctrl data. Adverse action 1: actions/events not recorded (audited) Adverse action 2: lost connectivity with syslog server
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. Threat agent: Attacker. Asset at risk: Auth, crypto, ctrl data. Adverse action 1: plaintext communications Adverse action 2: capturing network traffic; replay/man-in-the-middle
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. Threat agent: Attacker. Asset at risk: Auth, crypto, ctrl data. Adverse action 1: unpatched software version Adverse action 2: unknown update
T.USER_DATA_REUSE	User data may be inadvertently sent to a destination not intended by the original sender. Threat agent: Attacker. Asset at risk: Auth, crypto, ctrl data. Adverse action 1: sensitive data compromised Adverse action 2: network traffic re-use

530 **Table 11: Threats**

3.6 Organizational security policies

Organizational security policies (OSPs) are security rules, procedures, or guidelines enforced by the TOE, its operational environment, or a combination of the two.

3.6.1 OSPs enforced by TOE

The following security rules, procedures, or guidelines are enforced by the TOE.

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

Table 12: OSPs enforced by TOE

4 Security objectives

The security objectives are a concise and abstract statement of the intended solution to the security problem defined by the SPD.

4.1 Security objectives for the TOE

The security objectives for the TOE consists of a set of objectives the TOE should achieve to solve its part of the security problem.

O.PROTECTED_COMMUNICATIONS	The TOE will provide protected communication channels for administrators, other parts of a distributed TOE, and authorized IT entities
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.
O.SYSTEM_MONITORING	The TOE will provide the capability to generate audit data and send those data to an external IT entity.
O.DISPLAY_BANNER	The TOE will display an advisory warning regarding use of the TOE.
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.RESIDUAL_INFORMATION_CLEARING	The TOE will ensure that any data contained in a protected resource is not available when the resource is reallocated.
O.RESOURCE_AVAILABILITY	The TOE shall provide mechanisms that mitigate user attempts to exhaust TOE resources (e.g., persistent storage).
O.SESSION_LOCK	The TOE shall provide mechanisms that mitigate the risk of unattended sessions being hijacked.
O.TSF_SELF_TEST	The TOE will provide the capability to test some subset of its security functionality to ensure it is operating properly.

Table 13: Security objectives for the TOE

4.2 Security objectives for the environment

The security objectives for the environment consist of a set of objectives the environment should achieve to assist the TOE in correctly providing its security objectives.

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.

Table 14: Security objectives for the environment

4.3 Security objectives rationale

The security objectives rationale shows how the security objectives correspond to assumptions, threats, and organizational security policies and provide a justification of that tracing.

4.3.1 Tracing of security objectives to SPD

The tracing shows how the security objectives O.* and OE.* trace back to assumptions A.*, threats T.*, and organizational security policies OSP.* defined by the SPD.

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

Table 15: Tracing of security objectives to SPD

4.3.2 Justification of tracing

The justification demonstrates that the tracing of the security objectives to assumptions, threats, and OSPs is effective and all the given assumptions are upheld, all the given threats are countered, and all the given OSPs are enforced.

560 ### 4.3.2.1 Tracing of assumptions

Environment Objective	Rationale
OE.NO_GENERAL_PURPOSE	This security objective is necessary to address the assumption A.NO_GENERAL_PURPOSE by ensuring there are no general-purpose computing capabilities (e.g., the ability to execute arbitrary code or applications) capabilities on the TOE.
OE.PHYSICAL	This security objective is necessary to address the assumption A.PHYSICAL by ensuring the TOE and the data it contains is physically protected from unauthorized access.
OE.TRUSTED_ADMIN	This security objective is necessary to address the assumption A.TRUSTED_ADMIN by ensuring the administrators are non-hostile and follow all administrator guidance

Table 16: Assumption Rationale

4.3.2.2 Tracing of threats and OSPs

Objective	Rationale
O.PROTECTED_COMMUNICATIONS	This security objective is necessary to counter the threat: T.UNAUTHORIZED_ACCESS and T.UNAUTHORIZED_UPDATE to ensure the communications with the TOE is not compromised
O.VERIFIABLE_UPDATES	This security objective is necessary to counter the threat T.UNAUTHORIZED_UPDATE to ensure the end user has not installed a malicious update, thinking that it was legitimate.
O.SYSTEM_MONITORING	This security objective is necessary to counter the T.UNDETECTED_ACTIONS to ensure activity is monitored so the security of the TOE is not compromised.
O.DISPLAY_BANNER	This security objective is necessary to address the Organization Security Policy P.ACCESS_BANNER to ensure an advisory notice and consent warning message regarding unauthorized use of the TOE is displayed before the session is established.
O.TOE_ADMINISTRATION	This security objective is necessary to counter the T.ADMIN_ERROR that ensures actions performed on the TOE are logged so that indications of a failure or compromise of a TOE security mechanism are known and corrective actions can be taken.
O.RESIDUAL_INFORMATION_CLEARING	This security objective is necessary to counter the threat T.USER_DATA_REUSE so that data traversing the TOE could inadvertently be sent to a user other than that intended by the sender of the original network traffic.
O.RESOURCE_AVAILABILITY	This security objective is necessary to counter the threat: T.RESOURCE_EXHAUSTION to mitigate a denial of service, thus ensuring resources are available.
O.SESSION_LOCK	This security objective is necessary to counter the threat: T.UNAUTHORIZED_ACCESS to ensure accounts cannot be compromised and used by an attacker that does not otherwise have access to the TOE.
O.TSF_SELF_TEST	This security objective is necessary to counter the threat T.TSF_FAILURE to ensure failure of mechanisms do not lead to a compromise in the TSF.

Table 17: Threat and OSP Rationale

4.3.3 Security objectives conclusion

The tracing of the security objectives to assumptions, threats, and OSPs, and the justification of that tracing showed that all the given assumptions are upheld, all the given threats are countered, all the given OSPs are enforced, and the security problem as defined in the SPD is solved.

5 Extended components definition

The extended components definition specifies additional functional requirements not contained in CC Part 2 [CC-P2] or additional assurance requirements not contained in CC Part 3 [CC-P3].

The Extended SFRs are identified by having a label '_EXT' after the requirement name for TOE SFRs. The structure of the extended SFRs is modeled after the SFRs included in CC Part 2. The structure is as follows:

A. Class – The extended SFRs included in this ST are part of the identified classes of requirements.

B. Family – The extended SFRs included in this ST are part of several SFR families

C. Component – The extended SFRs are not hierarchical to any other components, though they may have identifiers terminating on other than "1". The dependencies for each extended component are identified in the TOE SFR Dependencies section of this ST.

D. The management requirements, if any, associated with the extended SFRs are incorporated into the Security management SFRs defined in this ST.

E. The audit requirements, if any, associated with the extended SFRs are incorporated into the Security audit SFRs defined in this ST.

F. The dependency requirements, if any, associated with the extended SFRs are identified in the dependency rationale and mapping section of the ST (Fulfillment of SFR dependencies).

5.1 Extended functional components

Additionally specified classes, families, and components not contained in [CC_P2].

5.1.1 FAU_STG_EXT.1 External Audit Trail Storage

FAU_STG_EXT.1.1 The TSF shall be able to [selection: transmit the generated audit data to an external IT entity over a trusted channel defined in FTP_ITC.1, receive and store audit data from an external IT entity over a trusted channel defined in FTP_ITC.1].

Application note: ECD-01
If the "receive and store" option is chosen from the selection above, the ST author should include details of the TSF audit data storage capability.

5.1.2 FAU_STG_EXT.3 Action in case of Loss of Audit Server Connectivity

Note: _EXT added to component name FAU_STG.3 below.

FAU_STG_EXT.3.1 The TSF shall [assignment: *action*] if the link to the external IT entity collecting the audit data generated by the TOE is not available.

Application note: ECD-02
The ST author fills in the action the TOE takes (pages the administrator, stops passing packets) if a link to the audit server is unavailable.

5.1.3 FCS_CKM_EXT.4 Cryptographic Key Zeroization

FCS_CKM_EXT.4.1 The TSF shall zeroize all plaintext secret and private cryptographic keys and CSPs when no longer required.

Application note: ECD-03
"Cryptographic Critical Security Parameters" are defined in FIPS 140-2 as "security-related information (e.g., secret and private cryptographic keys, and authentication data such as passwords and PINs) whose disclosure or modification can compromise the security of a cryptographic module." The zeroization indicated above applies to each intermediate storage area for plaintext key/cryptographic critical security parameter (i.e., any storage, such as memory buffers, that is included in the path of such data) upon the transfer of the key/cryptographic critical security parameter to another location.

5.1.4 FCS_COMM_PROT_EXT.1 Communications Protection

FCS_COMM_PROT_EXT.1.1 The TSF shall protect communications using [selection: IPSec, SSH] and [selection: TLS/HTTPS, no other protocol].

Application note: ECD-04
The intent of the above requirement is to use a cryptographic protocol to protect communications. Either IPSec or SSH is required; however, both may be selected if implemented by a conformant TOE. Additionally, TLS/HTTPS may be selected if that is implemented. After the ST author has made the appropriate selections, they are to select the detailed requirements in Annex C corresponding to their selection to put in the ST. As the assurance activities are associated with the specific protocols, this component has no associated assurance activities.

5.1.5 FCS_IPSEC_EXT.1 Explicit IPSEC

Note: From NDPP Annex C.

FCS_IPSEC_EXT.1.1 The TSF shall implement IPSec using the ESP protocol as defined by RFC 4303 using the cryptographic algorithms AES-CBC-128, AES-CBC-256 (both specified by RFC 3602), [selection: *no other algorithms, AES-GCM-128 as specified in RFC 4106, AES-GCM-256 as specified in RFC 4106*] and using IKEv1 as defined in RFCs 2407, 2408, 2409, and RFC 4109; [selection: *no other method, IKEv2 as defined in RFCs 4306, 4307*] to establish the security association.

Application note ECD-06
In subsequent publications of this PP, it is likely that AES-GCM will be required and CBC will become optional. Similarly, support for IKEv2 will likely be required, while support IKEv1 will become optional. Support for AES-CBC-128 and AES-CBC-256 is required above; if AES-GCM-128 or AES-GCM-256 are supported then the appropriate selection should be made, otherwise select "no other algorithm". It is acceptable to refine this requirement for IKEv1 and/or IKEv2 to include RFC 4868 as optional claimed hash algorithms. If this is done, the ST author should adjust FCS_COP.1(3) accordingly. Support for IKEv1 is required above; if IKEv2 is supported then that selection should be made, otherwise select "no other method." The ST author must make the appropriate selections and assignments to reflect the IPSec implementation. The ST author must provide enough detail to determine how the implementation is complying with the standard(s) identified; this can be done either by adding elements to this component, or by additional detail in the TSS. HMAC-SHA 1 is required by the RFCs as the hash algorithm used by the IKE implementation for CBC mode. If other hash algorithms are to be claimed, then either the requirement or the TSS section must identify those algorithms and the appropriate selections need to be made in FCS_COP.1(4).For IKEv1, the above requirement is to be interpreted as requiring the IKE implementation conforming to RFC 2409 with the additions/modifications as described in RFC 4109. Suite B algorithms (RFC 4869) are the preferred algorithms for implementation.

FCS_IPSEC_EXT.1.2 The TSF shall ensure that IKEv1 Phase 1 exchanges use only main mode.

FCS_IPSEC_EXT.1.3 The TSF shall ensure that IKEv1 SA lifetimes are able to be limited to 24 hours for Phase 1 SAs and 8 hours for Phase 2 SAs.

Application note: ECD-07
The above requirement can be accomplished either by providing Security Administrator-configurable lifetimes (with appropriate FMT requirements and instructions in documents mandated by AGD_OPE, as necessary), or by "hard coding" the limits in the implementation.

FCS_IPSEC_EXT.1.4 The TSF shall ensure that IKEv1 SA lifetimes are able to be limited to [assignment: *number between 100 - 200*] MB of traffic for Phase 2 SAs.

Application note: ECD-08
The above requirement can be accomplished either by providing Security Administrator-configurable lifetimes (with appropriate FMT requirements and instructions in documents mandated by AGD_OPE), or by "hard coding" the limits in the implementation. The ST author selects the amount of data in the range specified by the requirement. In general, instructions for setting the parameters of the implementation, including lifetime of the SAs, should be specified through FMT requirements and included in the administrative guidance generated for AGD_OPE.

FCS_IPSEC_EXT.1.5 The TSF shall ensure that all IKE protocols implement DH Groups 14 (2048-bit MODP), and [selection: *24 (2048-bit MODP with 256-bit POS), 19 (256-bit Random ECP), 20 (384-bit Random ECP)*, [assignment: *other DH groups that are implemented by the TOE*], *no other DH groups*].

Application Not: ECD-09
The above requires that the TOE support DH Group 14. If other groups are supported, then those should be selected (for groups 24, 19, and 20) or specified in the assignment above; otherwise "no other DH groups" should be selected. This applies to IKEv1 and (if implemented) IKEv2 exchanges. In future publications of this PP DH Groups 19 (256-bit Random ECP) and 20 (384-bit RandomECP) will be required.

FCS_IPSEC_EXT.1.6 The TSF shall ensure that all IKE protocols implement Peer Authentication using the [selection: *DSA, rDSA, ECDSA*] algorithm.

Application note: ECD-10
The selected algorithm should correspond to an appropriate selection for FCS_COP.1(2).

FCS_IPSEC_EXT.1.7 The TSF shall support the use of pre-shared keys (as referenced in the RFCs) for use in authenticating its IPSec connections.

FCS_IPSEC_EXT.1.8 The TSF shall support the following:
1. Pre-shared keys shall be able to be composed of any combination of upper and lower case letters, numbers, and special characters (that include: "!", "@", "#", "$", "%", "^", "&", "", "(", and ")");*
2. Pre-shared keys of 22 characters and [selection: [assignment: other supported lengths], no other lengths].

Application note: ECD-11
For the length of the pre-shared keys, a common length (22 characters) is required to help promote interoperability. If other lengths are supported they should be listed in the assignment; this assignment can also specify a range of values (e.g., "lengths from 5 to 55 characters") as well.

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

Note: ( ) removed from component name FCS_RBG_(EXT).*

FCS_RBG_EXT.1.1 The TSF shall perform all random bit generation (RBG) services in accordance with [selection, choose one of: NIST Special Publication 800-90 using [selection: Hash_DRBG (any), HMAC_DRBG (any), CTR_DRBG (AES), Dual_EC_DRBG (any)]; FIPS Pub 140-2 Annex C: X9.31 Appendix 2.4 using AES] seeded by an entropy source that accumulated entropy from at least one independent TSF-hardware-based noise source.

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded with a minimum of [selection, choose one of: 128 bits, 256 bits] of entropy at least equal to the greatest bit length of the keys and authorization factors that it will generate.

Application note ECD-12
NIST Special Pub 800-90, Appendix C describes the minimum entropy measurement that will probably be required future versions of FIPS-140. If possible this should be used immediately and will be required in future versions of this PP. For the first selection in FCS_RBG_(EXT).1.1, the ST author should select the standard to which the RBG services comply (either 800-90 or 140-2 Annex C). SP 800-90 contains four different methods of generating random numbers; each of these, in turn, depends on underlying cryptographic primitives (hash functions/ciphers). The ST author will select the function used (if 800-90 is selected), and include the specific underlying cryptographic primitives used in the requirement or in the TSS. While any of the identified hash functions (SHA-1, SHA-224, SHA-256, SHA-384, SHA-512) are allowed for Hash_DRBG or HMAC_DRBG, only AES-based implementations for CT_DRBG are allowed. While any of the curves defined in 800-90 are allowed for Dual_EC_DRBG, the ST author not only must include the curve chosen, but also the hash algorithm used. Note that for FIPS Pub 140-2 Annex C, currently only the method described 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 is valid. If the key length for the AES implementation used here is different than that used to encrypt the user data, then FCS_COP.1 may have to be adjusted or iterated to reflect the different key length. For the selection in FCS_RBG_(EXT).1.2, the ST author selects the minimum number of bits of entropy that is used to seed the RBG. The ST author also ensures that any underlying functions are included in the baseline requirements for the TOE.

5.1.7 FIA_PMG_EXT.1 Password Management

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 special characters (that include: “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, and “)”);
2. Minimum password length shall settable by the Security Administrator, and support passwords of 8 characters or greater;
3. Passwords composition rules specifying the types and number of required characters that comprise the password shall be settable by the Security Administrator.

Application note: ECD-19
The intent of this caveat is that the Security Administrator is able to specify, for example, that passwords contain at least 1 upper case letter, 1 lower case letter, 1

numeric character, and 1 special character, and the TOE enforces this restriction. “Types” refers to all of the types listed in item 1 in this element.

4. Passwords shall have a maximum lifetime, configurable by the Security Administrator.
5. New passwords must contain a minimum of 4 character changes from the revious password.

Application note: ECD-20
Note that it is not necessary to store a plaintext version of the password in order to determine that at least 4 characters have changed, since FIA_UAU.6 requires re-authentication when changing the password. "Administrative passwords" refers to passwords used by administrators at the local console or over protocols that support passwords, such as SSH and HTTPS.

5.1.8 FIA_UAU_EXT.5 Extended: Password-based Authentication Mechanism

FIA_UAU_EXT.5.1 The TSF shall provide a local password-based authentication mechanism, [selection: [assignment: *other authentication mechanism(s)*], none] to perform user authentication.

FIA_UAU_EXT.5.2 The TSF shall ensure that users with expired passwords are [selection: required to create a new password after correctly entering the expired password, locked out until their password is reset by an administrator].

Application note: ECD-21
The ST author can fill in the assignment with any other supported authentication mechanisms that are not local, such as a RADIUS server. If no external authentication mechanisms are supported, the ST author should choose "none" in the selection.

5.1.9 FIA_UIA_EXT.1 User Identification and Authentication

FIA_UIA_EXT.1.1 The TSF shall allow [selection:[assignment: *list of TOE-provided services*], *no services*] on behalf of the user to be performed before the user is identified and authenticated.

FIA_UIA_EXT.1.2 The TSF shall require each user to be successfully identified and authenticated before allowing any other TSF-mediated actions on behalf of that user.

Application note: ECD-22
This requirement applies to users (administrators) of services available from the TOE directly, and not services available by connecting through the TOE. Authentication can be password-based through the local console or through a protocol that supports passwords (such as SSH), or be certificate based (SSH, TLS).

Note: Error in NDPP: the following component FPT_PTD.1 should read FPT_PTD_EXT.1, because FPT_PTD.1 is NOT contained in CC Part 2.

5.1.10 FPT_PTD.1(1) Management of TSF Data (for reading of authentication data)

FPT_PTD.1.1(1) **Refinement:** The TSF shall **prevent** *reading of* the *plaintext passwords.*

Application note ECD-23
The intent of the requirement is that no user or administrator be able to read the authentication data used to directly authenticate a user to the TSF (such as an unencrypted password) through “normal” interfaces if the reading of such data

could lead to someone impersonating that user. An all-powerful administrator of course could directly read memory to capture a password but is trusted not to do so. Likewise, if a system relies on a public key for a user as part of the authentication process, that key could be considered “authentication data” but being able to read that key would not lead to a compromise of that user, and so would not fall under the purview of this requirement.

5.1.11 FPT_PTD.1(2) Management of TSF Data (for reading of all symmetric keys)

FPT_PTD.1.1(2) **Refinement:** The TSF shall **prevent** *reading of all pre-shared keys, symmetric key, and private keys.*

Application note: ECD-24
The intent of the requirement is that no user or administrator be able to read or view the identified keys (stored or ephemeral) through “normal” interfaces. While the security administrator of course could directly read memory to view these keys, they are trusted not to do so.

5.1.12 FPT_TST_EXT.1: TSF Testing

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.

5.1.13 FPT_TUD_EXT.1 Extended: Trusted Update

Note: ( ) removed from component name FPT_TUD_(EXT).*

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 [selection: digital signature mechanism, published hash] prior to installing those updates.

Application note: ECD-25
The digital signature mechanism referenced in the third element is the one specified in FCS_COP.1(2). The published hash referenced is generated by one of the functions specified in FCS_COP.1(3). In subsequent publications of this PP, it is likely that digital signatures will be required.

5.1.14 FTA_SSL_EXT.1 TSF-initiated Session Locking

FTA_SSL_EXT.1.1 The TSF shall, for local interactive sessions, [selection: lock the session - disable any activity of the user’s data access/display devices other than unlocking the session, and requiring that the administrator re-authenticate to the TSF prior to unlocking the session; terminate the session] after a Security Administrator-specified time period of inactivity.

5.2 Extended assurance components

None specified.

6 Security requirements

Security requirements specify the security objectives of the TOE in a standardized manner. Security requirements fall into two groups: Security functional requirements (SFRs) and security assurance requirements (SARs).

SFRs are modeled by using components from CC Part 2 [CC-P2]. SARs are modeled by using components from CC Part 3 [CC-P3]. Necessary modifications to SFRs and SARs are performed through the permitted operations assignment, selection, iteration and refinement. This document uses the following font conventions to identify the operations defined by the CC:

- 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***]).
- Iteration: allows a component to be used more than once with varying operations. In the ST, iteration is indicated by a number placed at the end of the component. For example FDP_IFF.1(1) and FDP_IFF.1(2) indicate that the ST includes two iterations of the FDP_IFF.1 requirement, (1) and (2).
- 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 …").
- Extended Requirements (i.e., those not found in Part 2 of the CC) are identified with "_EXT" in the functional class/name.
- Where operations were completed in the NDPP itself, the formatting has been updated to follow the formatting as described above for consistency with all operations performed.

Other sections of the ST use bolding to highlight text of special interest, such as captions.

6.1 Security functional requirements

The SFRs are a translation of the security objectives of the TOE and are independent of any technical implementation.

6.1.1 Overview of SFRs

This section identifies the Security Functional Requirements for the TOE. The Security Functional Requirements included in this section are derived from *US Government, Security Requirements for Network Devices (pp_nd_v1.0), version 1.0, dated 10 December 2011* and Part 2 of the *Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 3, dated: July 2009* and all international interpretations.

Functional Component	
Requirement Class	**Requirement Component**
FAU: Security audit	FAU_GEN.1: Audit data generation
	FAU_GEN.2: User identity association
	FAU_STG_EXT.1: External audit trail storage
	FAU_STG_EXT.3: Action in case of loss of audit server connectivity
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)

Functional Component	
	FCS_COP.1(2): Cryptographic operation (for cryptographic signature)
	FCS_COP.1(3): Cryptographic operation (for cryptographic hashing)
	FCS_COP.1(4): Cryptographic operation (for keyed-hash message authentication)
	FCS_RBG_EXT.1: Cryptographic operation (random bit generation)
	FCS_COMM_PROT_EXT.1: Communications protection
	FCS_IPSEC_EXT.1: IPSEC
FDP: User data protection	FDP_RIP.2: Full residual information protection
FIA: Identification and authentication	FIA_PMG_EXT.1: Password management
	FIA_UIA_EXT.1: User identification and authentication
	FIA_UAU_EXT.5: Password-based authentication mechanism
	FIA_UAU.6: Re-authenticating
	FIA_UAU.7: Protected authentication feedback
FMT: Security management	FMT_MTD.1: Management of TSF data (for general TSF data)
	FMT_SMF.1: Specification of management functions
	FMT_SMR.1: Security roles
FPT: Protection of the TSF	FPT_ITT.1(1): Basic internal TSF data transfer protection (disclosure)
	FPT_ITT.1(2): Basic internal TSF data transfer protection (modification)
	FPT_PTD_EXT.1(1): Management of TSF data (for reading of authentication data)
	FPT_PTD_EXT.1(2): Management of TSF data (for reading of keys)
	FPT_RPL.1: Replay detection
	FPT_STM.1: Reliable time stamps
	FPT_TUD_EXT.1: Trusted update
	FPT_TST_EXT.1: TSF testing
FRU: Resource utilization	FRU_RSA.1: Maximum quotas
FTA: TOE Access	FTA_SSL_EXT.1: TSF-initiated session locking
	FTA_SSL.3: TSF-initiated termination
	FTA_TAB.1: Default ToE access banners

Functional Component	
FTP: Trusted path/channels	FTP_ITC.1(1): Inter-TSF trusted channel (prevention of disclosure)
	FTP_ITC.1(2): Inter-TSF trusted channel (detection of modification)
	FTP_TRP.1(1): Trusted path (prevention of disclosure)
	FTP_TRP.1(2): Trusted path (detection of modification)

Table 18: Overview of SFRs

880 ### 6.1.2 FAU: Security audit

The CC class security audit involves recognizing, recording, storing, and analyzing information related to security relevant activities controlled by the TOE.

FAU_GEN.1	**Audit data generation**
Hierarchical to:	No other components.
Dependencies:	FPT_STM.1 Reliable time stamps
FAU_GEN.1.1	The TSF shall be able to generate an audit record of the following auditable events: a) Start-up and shutdown of the audit functions; b) All auditable events for the [***basic***] level of audit; and c) [**All administrative actions;** d) **Specifically defined auditable events listed in Table 19: Auditable Events:**].
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 (if applicable), 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 19: Auditable Events**]

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None.	
FAU_GEN.2	None.	
FAU_STG_EXT.1	None.	
FAU_STG_EXT.3	Loss of connectivity.	No additional information.
FCS_CKM.1	Failure on invoking functionality.	No additional information.
FCS_CKM_EXT.4	Failure on invoking functionality.	No additional information.
FCS_COP.1(1)	Failure on invoking functionality.	No additional information.
FCS_COP.1(2)	Failure on invoking functionality.	No additional information.

Requirement	Auditable Events	Additional Audit Record Contents
FCS_COP.1(3)	Failure on invoking functionality.	No additional information.
FCS_COP.1(4)	Failure on invoking functionality.	No additional information.
FCS_RBG_EXT.1	Failure of the randomization process.	No additional information.
FCS_COMM_PROT_EXT.1	None.	
FCS_IPSEC_EXT.1	Failure to establish an IPSec SA. Establishment/Termination of an IPSec SA.	Reason for failure. Non-ToE endpoint of connection (IP address) for both successes and failures.
FDP_RIP.2	None.	
FIA_PMG_EXT.1	None.	
FIA_UIA_EXT.1	All use of the identification and authentication mechanism.	Provided user identity, origin of the attempt (e.g., IP address).
FIA_UAU_EXT.5	All use of the authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UAU.6	Attempt to re-authenticate.	Origin of the attempt (e.g., IP address).
FIA_UAU.7	None.	
FMT_MTD.1	None.	
FMT_SMF.1	None.	
FMT_SMR.1	None.	
FPT_ITT.1(1)	None.	
FPT_ITT.1(2)	None.	
FPT_PTD_EXT.1(1)	None.	
FPT_PTD_EXT.1(2)	None.	
FPT_RPL.1	Detected replay attacks.	Origin of the attempt (e.g., IP address).
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	Indication that TSF self-test was completed.	Any additional information generated by the tests beyond “success” or “failure”.
FRU_RSA.1	Maximum quota being exceeded.	Resource identifier.
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_TAB.1	None.	

Requirement	Auditable Events	Additional Audit Record Contents
FTP_ITC.1(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_ITC.1(2)	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(1)	Initiation of the trusted channel. Termination of the trusted channel. Failures of the trusted path functions.	Identification of the claimed user identity.
FTP_TRP.1(2)	Initiation of the trusted channel. Termination of the trusted channel. Failures of the trusted path functions.	Identification of the claimed user identity.

Table 19: Auditable Events

FAU_GEN.2	**User identity association**
Hierarchical to:	No other components.
Dependencies:	FAU_GEN.1 Audit data generation FIA_UID.1 Timing of identification
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.

FAU_STG_EXT.1	**External audit trail storage**
Hierarchical to:	No other components.
Dependencies:	FAU_GEN.1 Audit data generation FCS_IPSEC_EXT.1 IPSec
FAU_STG_EXT.1.1	The TSF shall be able to [***transmit the generated audit data to an external IT entity using IPSec as defined in FCS_IPSEC_EXT.1***].

890

FAU_STG_EXT.3	**Action in case of Loss of Audit Server Connectivity**
Hierarchical to:	No other components.

Dependencies:	No dependencies.
FAU_STG_EXT.3.1	The TSF shall [**queue audit records on the TOE and attempt re-establish connection**] if the link to the external IT entity collecting the audit data generated by the ToE is not available.

6.1.3 FCS: Cryptographic support

The CC class cryptographic support provides functionality to support cryptographic functions implemented by the TOE.

FCS_CKM.1	**Cryptographic key generation**
Hierarchical to:	No other components.
Dependencies:	FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation, FCS_CKM.4 Cryptographic key destruction
FCS_CKM.1.1	The TSF shall generate **asymmetric** cryptographic **keys in accordance with a domain parameter generator and** [***a random number generator***] that meet the following: **a) All cases: (i.e., any of the above)** • **ANSI X9.80 (3 January 2000), "Prime Number Generation, Primality Testing, and Primality Certificates" using random integers with deterministic tests, or constructive generation methods** • **Generated key strength shall be equivalent to, or greater than, a symmetric key strength of 112 bits using conservative estimates.** ~~**b) Case: For domain parameters used in finite field-based key establishment schemes**~~ ~~• **NIST Special Publication 800-56A "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"**~~ **c) Case: For domain parameters used in RSA-based key establishment schemes** • **NIST Special Publication 800-56B "Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography"** ~~**d) Case: For domain parameters used in elliptic curve-based key establishment schemes**~~ ~~• **NIST Special Publication 800-56A "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"**~~ ~~• **The TSF shall implement "NIST curves" P-256, P-384 and [Selection: P-521, no other curves] (as defined in FIPS PUB 186-3, "Digital Signature Standard")**~~

Application note:
The key generation as required in FCS_CKM.1 meets also the newer ANSI X9.80 (2005) version for "Prime Number Generation, Primality Testing, and Primality Certificates".

900

FCS_CKM_EXT.4	**Cryptographic key destruction**
Hierarchical to:	No other components.
Dependencies:	FCS_CKM.1 Cryptographic key generation
FCS_CKM_EXT.4.1	The TSF shall zeroize all plaintext secret and private cryptographic keys and CSPs when no longer required.

FCS_COP.1(1)	**Cryptographic operation (for data encryption/decryption)**
Hierarchical to:	No other components.
Dependencies:	FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation FCS_CKM.4 Cryptographic Key destruction
FCS_COP.1(1).1	The TSF shall perform [**encryption and decryption**] in accordance with a specified cryptographic algorithm [**AES operating in** [**CBC and GCM mode**]] and cryptographic key sizes [**128-bits, 256-bits**]**, and** [***192 bits***] that meet**s** the following: • **FIPS PUB 197, "Advanced Encryption Standard (AES)"** • [***NIST SP 800-38A, NIST SP 800-38D***].

FCS_COP.1(2)	**Cryptographic operation (for cryptographic signature)**
Hierarchical to:	No other components.
Dependencies:	FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation FCS_CKM.4 Cryptographic Key destruction
FCS_COP.1(2).1	The TSF shall perform [**cryptographic signature services**] in accordance with a ~~specified cryptographic algorithm~~ [***(2) RSA Digital Signature Algorithm (rDSA)*** ~~*and crytpgraphic*~~ ***with a*** *key size*~~*s*~~ ***(modulus) of 2048 bits or greater***] that meet**s** the following: **Case: RSA Digital Signature Algorithm** • [***FIPS PUB 186-3, "Digital Signature Standard"***].

FCS_COP.1(3)	**Cryptographic operation (for cryptographic hashing)**
Hierarchical to:	No other components.
Dependencies:	FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation FCS_CKM.4 Cryptographic Key destruction
FCS_COP.1(3).1	The TSF shall perform [**cryptographic hashing services**] in accordance with a specified cryptographic algorithm [***SHA-1, SHA-256, SHA-384, SHA-512***] and ~~cryptographic key sizes~~ **message digest sizes** [***160, 256, 384, 512***] **bits** that meet the following: [**FIPS Pub 180-3 "Secure Hash Standard."**].

FCS_COP.1(4)	**Cryptographic operation (for keyed-hash message authentication)**
Hierarchical to:	No other components.
Dependencies:	FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation FCS_CKM.4 Cryptographic Key destruction
FCS_COP.1(4).1	The TSF shall perform [**keyed-hash message authentication**] in accordance with a specified cryptographic algorithm **HMAC-**[***SHA-1, SHA-256, SHA-512***]**,** ~~and cryptographic~~ **key size**~~s~~ [***128, 192, 256 bits***], **and message digest sizes** [***160, 256, 512***] **bits** that meet the following:[***FIPS Pub 198-1 "The Keyed-Hash Message Authentication Code", and FIPS PUB 180-3, "Secure Hash Standard."***].

FCS_RBG_EXT.1	**Cryptographic operation (random bit generation)**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FCS_RBG_EXT.1.1	The TSF shall perform all random bit generation (RBG) services in accordance with [***NIST Special Publication 800-90 using CTR_DRBG (AES)***] seeded by an entropy source that accumulated entropy from at least one independent TSF-hardware-based noise source.
FCS_RBG_EXT.1.2	The deterministic RBG shall be seeded with a minimum of [***256 bits***] of entropy at least equal to the greatest length of the keys and authorization factors that it will generate.

910 **Application note:**
FCS_RBG_EXT.1.2 means that the seeding process for the deterministic RBG uses an entropy input with a minimum of 256 bits of entropy coming from the TSF-hardware-based noise source. During the seeding process both the instantiation and the reseeding functions from [NIST Special Publication 800-90 using CTR_DRBG (AES)] call the conditioning function "Block_Cipher_df". This function iterates a CBC-MAC calculation with a 128 bit output block on the entropy input. Therefore the DRNG internal state after instantiation or reseeding has a min-entropy not exceeding 128 bits.

In addition to this upper bound on the min-entropy of the internal DRNG state the following application note specifies the lower bound in DRG.2.1.

920
Application note:
According to [KS2011] the TSF provides a deterministic random number generator of the pre-defined class DRG.2 that implements the following security capabilities:

DRG.2.1 If initialized with a random seed using a PTRNG of class PTG.2 as random source together with a nonce and a personalization string, the internal state of the RNG shall have a min-entropy of at least 80 bits.
DRG.2.2 The RNG provides forward secrecy.
DRG.2.3 The RNG provides backward secrecy.

The TSF provides random numbers that meet:

DRG.2.4 The RNG initialized with a random seed of 384 bits at startup generates output for which 2^{14} strings of bit length 128 are mutually different with probability of greater than $(1-2^{-8})$.
DRG.2.5 Statistical test suites cannot practically distinguish the random numbers from output sequences of an ideal RNG. The random numbers pass test procedure A (see [KS2011]).

FCS_COMM_PROT_EXT.1	**Communications protection**
Hierarchical to:	No other components.
Dependencies:	FCS_IPSEC_EXT.1 IPSec, or FCS_SSH_EXT.1 SSH
FCS_COMM_PROT_EXT.1.1	The TSF shall protect communications using [***IPSec***] and [***no other protocol***].

FCS_IPSEC_EXT.1	**IPSEC**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FCS_IPSEC_EXT.1.1	The TSF shall implement IPSec using the ESP protocol as defined by RFC 4303 using the cryptographic algorithms AES-CBC-128, AES-CBC-256 (both specified by RFC 3602), [***no other algorithms***] and using IKEv1 as defined in RFCs 2407, 2408, 2409, and RFC 4109, [***no other methods***] to establish the security association.

FCS_IPSEC_EXT.1.2	The TSF shall ensure that IKEv1 Phase 1 exchanges use only main mode.
FCS_IPSEC_EXT.1.3	The TSF shall ensure that IKEv1 SA lifetimes are able to be limited to 24 hours for Phase 1 SAs and 8 hours for Phase 2 SAs.
FCS_IPSEC_EXT.1.4	The TSF shall ensure that IKEv1 SA lifetimes are able to be limited to [**an administratively configurable number of kilobytes including the range from 100 – 200**] MB of traffic for Phase 2 SAs.
FCS_IPSEC_EXT.1.5	The TSF shall ensure that all IKE protocols implement DH Groups 14 (2048-bit MODP) and [**no other DH groups**].
FCS_IPSEC_EXT.1.6	The TSF shall ensure that all IKE protocols implement Peer Authentication using the [***rDSA***] algorithm.
FCS_IPSEC_EXT.1.7	The TSF shall support the use of pre-shared keys (as referenced in the RFCs) for use in authenticating its IPSec connections.
FCS_IPSEC_EXT.1.8	The TSF shall support the following: • Pre-shared keys shall be able to be composed of any combination of upper and lower case letters, numbers, and special characters (that include: “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”,**“”” (double quote), “’” (single quote), “+”, “,” (comma), “-”, “.”, “/”, “:”, “;”, “<”, “=”, “>”, “[”, “\”, “]”, “_”, “`” (apostrophe), and “~”**; • Pre-shared keys of 22 characters [**[*up to 128 characters*]**].

930

6.1.4 FDP: User data protection

The CC class user data protection provides functionality to protect user data during import to the TOE, storage by the TOE, or export from the TOE.

FDP_RIP.2	**Full residual information protection**
Hierarchical to:	FDP_RIP.1 Subset residual information protection
Dependencies:	No dependencies.
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.

940

6.1.5 FIA: Identification and authentication

The CC class identification and authentication addresses the requirements for functions to establish and verify a claimed user identity.

FIA_PMG_EXT.1	**Password management**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
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 special characters (that include: "!", "@", "#", "$", "%", "^", "&", "*", "(", and ")"); 2. Minimum password length shall be settable by the Security Administrator, and support passwords of 8 characters or greater; 3. Passwords composition rules specifying the types and number of required characters that comprise the password shall be settable by the Security Administrator. 4. Passwords shall have a maximum lifetime, configurable by the Security Administrator. 5. New passwords must contain a minimum of 4 character changes from the previous password.

FIA_UAU_EXT.5	**Password-based authentication mechanism**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FIA_UAU_EXT.5.1	The TSF shall provide a local password-based authentication mechanism, [**[*remote password-based authentication via RADIUS or TACACS+*]**] to perform user authentication.
FIA_UAU_EXT.5.2	The TSF shall ensure that users with expired passwords are [***required to create a new password after correctly entering the expired password***].

950

FIA_UAU.6	**Re-authenticating**

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FIA_UAU.6.1	The TSF shall re-authenticate the user under the conditions**: when the user changes their password,** [***following TSF-initiated locking (FTA_SSL),*** **[***no other conditions***]**].

FIA_UAU.7	**Protected authentication feedback**
Hierarchical to:	No other components.
Dependencies:	FIA_UAU.1 Timing of authentication
FIA_UAU.7.1	The TSF shall provide only [**obscured feedback**] to the user while the authentication is in progress **at the local console**.

FIA_UIA_EXT.1	**User identification and authentication**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FIA_UIA_EXT.1.1	The TSF shall allow [***no services***] on behalf of the user to be performed before the user is identified and authenticated.
FIA_UIA_EXT.1.2	The TSF shall require each user to be successfully identified and authenticated before allowing any other TSF-mediated actions on behalf of that user.

6.1.6 FMT: Security management

The CC class security management specifies the management of security attributes, TSF data and TSF functions.

960

FMT_MTD.1	**Management of TSF data (for general TSF data)**
Hierarchical to:	No other components.
Dependencies:	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions

FMT_MTD.1.1	The TSF shall restrict the ability to [**[*manage*]**] the [***TSF data***] to **the ~~Security Administrators~~ privileged administrator, and semi-privileged administrator with appropriate privileges**.

FMT_SMF.1	**Specification of Management Functions**
Hierarchical to:	No other components.
Dependencies:	No dependencies
FMT_SMF.1.1	The TSF shall be capable of performing the following management functions: [• **Ability to configure the list of TOE services available before an entity is identified and authenticated, as specified in FIA_UIA_EXT.1, respectively.** • **Ability to configure the cryptographic functionality.** • **Ability to update the TOE, and to verify the updates using the digital signature capability (FCS_COP.1(2)) and** [***no other functions***] • **Ability to manage the cryptographic functionality** • **Ability to manage the audit logs and functions** • **Ability to manage routing tables** • **Ability to manage security attributes belonging to individual users** • **Ability to manage the default values of the security attributes** • **Ability to manage the warning banner message and content** • **Ability to manage the time limits of session inactivity**].

FMT_SMR.1	**Security roles**
Hierarchical to:	No other components.
Dependencies:	FIA_UID.1 Timing of identification
FMT_SMR.1.1	The TSF shall maintain the **following** ~~roles~~ [**administrative privilege levels and non-administrative access** [**0, 1(administrator), 15 (privileged administrator), custom levels 2-14 (semi-privileged administrator), non-administrative access (neighbor routers)**].
FMT_SMR.1.2	The TSF shall be able to associate users with ~~roles~~ **administrative privilege levels and non-administrative access**.

6.1.7 FPT: Protection of the TSF

The CC class protection of the TSF provides functionality to protect TSF data during interaction with external entities, during interaction with administrative databases that guide the enforcement of the SFRs, or during execution of the implemented functions that enforce the SFRs.

970

FPT_ITT.1(1)	**Basic internal TSF data transfer protection (disclosure)**
Hierarchical to:	No other components.
Dependencies:	No dependencies
FPT_ITT.1(1).1	The TSF shall protect TSF data from [***disclosure***] when it is transmitted between separate parts of the TOE **through the use of the TSF-provided cryptographic services: [FCS_IPSEC_EXT.1 IPSEC]**.

FPT_ITT.1(2)	**Basic internal TSF data transfer protection (modification)**
Hierarchical to:	No other components.
Dependencies:	No dependencies
FPT_ITT.1(2).1	The TSF shall ~~protect~~ **detect modification of** TSF data ~~from~~ when it is transmitted between separate parts of the TOE **through the use of the TSF-provided cryptographic services: [FCS_IPSEC_EXT.1 IPSEC]**.

FPT_PTD_EXT.1(1)	**Management of TSF data (for reading of authentication data)**
Hierarchical to:	No other components.
Dependencies:	No dependencies
FPT_PTD_EXT.1(1).1	The TSF shall prevent reading of the plaintext passwords.

FPT_PTD_EXT.1(2)	**Management of TSF data (for reading of all symmetric keys)**
Hierarchical to:	No other components.
Dependencies:	FCS_CKM.1 Cryptographic key management
FPT_PTD_EXT.1(2).1	The TSF shall prevent reading of all pre-shared keys, symmetric key, and private keys.

FPT_RPL.1	**Replay detection**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FPT_RPL.1.1	The TSF shall detect replay for the following entities: [**network packets terminated at the TOE**].
FPT_RPL.1.2	The TSF shall perform: [**reject the data**] when replay is detected.

980

FPT_STM.1	**Reliable time stamps**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FPT_STM.1.1	The TSF shall be able to provide reliable time stamps for its own use.

FPT_TST_EXT.1	**TSF Testing**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
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.

FPT_TUD_EXT.1	**Trusted update**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
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 the TOE firmware/software.
FPT_TUD_EXT.1.3	The TSF shall provide a means to verify firmware/software updates to the TOE using a [***published hash***] prior to installing those updates.

6.1.8 FRU: Resource utilization

The CC class resource utilization provides functionality that support the availability of required resources, like processing capability or storage capacity.

FRU_RSA.1	**Maximum quotas**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FRU_RSA.1.1	The TSF shall enforce maximum quotas of the following resources: [**resources supporting the administrative interface**]**,** [**[*no other resource*]**] that [***individual user***] can use [***simultaneously***].

990

6.1.9 FTA: TOE access

The CC class TOE access specifies requirements for controlling the establishment of user sessions.

FTA_SSL_EXT.1	**TSF-initiated session locking**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FTA_SSL_EXT.1.1	The TSF shall, for local interactive sessions, [***terminate the session***] after a**n** ~~Security~~ Administrator-specified time period of inactivity.

FTA_SSL.3	**TSF-initiated termination**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FTA_SSL.3.1	The TSF shall terminate **a remote** ~~an~~ interactive session after a [**~~Security~~ Administrator-configurable time interval of session inactivity**].

FTA_TAB.1	**Default TOE access banners**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FTA_TAB.1.1	Before establishing a user **/administrator** session, the TSF shall display **an**

	~~**Security**~~ **Administrator-specified** advisory **notice and consent** warning message regarding unauthorized use of the TOE.

6.1.10 FTP: Trusted path/channels

The CC class trusted path/channels specifies requirements for a trusted communication path between users and the TSF or between other trusted IT products and the TSF.

FTP_ITC.1(1)	**Inter-TSF trusted channel (prevention of disclosure)**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FTP_ITC.1(1).1	The TSF shall **use IPsec as defined in FCS_IPSEC_EXT.1 with AES as defined in FCS_COP.1(1).1 to** provide a **trusted** communication channel between itself and ~~another trusted~~ **authorized** IT ~~product~~ **entities** that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from ~~modification or~~ disclosure.
FTP_ITC.1(1).2	The TSF shall permit [***the TSF***, **or the authorized IT entities**] to initiate communication via the trusted channel.
FTP_ITC.1(1).3	The TSF shall initiate communication via the trusted channel for [**syslog server (storage of audit records), *AAA (remote authentication) and NTP,* [AES as defined in FCS_COP.1(1).1]**].

1000

FTP_ITC.1(2)	**Inter-TSF trusted channel (detection of modification)**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FTP_ITC.1(2).1	The TSF shall **use** [**keyed hash as defined in FCS_COP.1(4).1**] **in** ~~provide~~ **providing** a **trusted** communication channel between itself and ~~another trusted~~ **authorized** IT ~~product~~ **entities** that is logically distinct from other communication channels and provides assured identification of its end points and ~~protection of the channel data from modification or disclosure~~ **detection of the modification of data**.
FTP_ITC.1(2).2	The TSF shall permit [***the TSF***, **or the authorized IT entities**] to initiate communication via the trusted channel.
FTP_ITC.1(2).3	The TSF shall initiate communication via the trusted channel for [**syslog server (storage of audit records),** [***AAA (remote authentication) and NTP,*** **[keyed hash as defined in FCS_COP.1(4).1]**].

FTP_TRP.1(1)	**Trusted path (prevention of disclosure)**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FTP_TRP.1(1).1	The TSF shall provide a communication path between itself and [***remote* administrators**] **using [IPSec as specified in FCS_IPSEC_EXT.1 to access the CLI]** ~~users~~ that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from [***disclosure***].
FTP_TRP.1(1).2	The TSF shall permit [***remote administrators***] to initiate communication via the trusted path.
FTP_TRP.1(1).3	The TSF shall require the use of the trusted path for [***all remote administrative actions***].

FTP_TRP.1(2)	**Trusted path (detection of modification)**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FTP_TRP.1(2).1	The TSF shall provide a communication path between itself and [***remote administrators***] **using [IPSec as specified in FCS_IPSEC_EXT.1 to access the CLI]** ~~users~~ that is logically distinct from other communication paths and provides assured identification of its end points and ~~protection~~ **detection of modification** of the communicated data ~~from~~.
FTP_TRP.1(2).2	The TSF shall permit [***remote administrators***] to initiate communication via the trusted path.
FTP_TRP.1(2).3	The TSF shall require the use of the trusted path for [***all remote administrative actions***].

6.2 Security assurance requirements

The SARs are a description of how the TOE is to be evaluated.

6.2.1 Overview of SARs

The TOE assurance requirements for this ST are taken directly from the NDPP which are derived from Common Criteria Version 3.1, Revision 3. The assurance requirements are summarized in the table below as identified in the NDPP, Section 4.3. The ST does not include any changes to the assurance

requirements beyond those identified and described in the NDPP, as such all assurance activities from NDPPv1.0 form the SARs in this ST.

ASE: CC assurance class security target evaluation	
ASE_CCL.1	Conformance claims
ASE_ECD.1	Extended components definition
ASE_INT.1	ST introduction
ASE_OBJ.2	Security objectives
ASE_REQ.2	Derived security requirements
ASE_SPD.1	Security problem definition
ASE_TSS.1	TOE summary specification
ALC: CC assurance class life cycle support	
ALC_CMC.1	Labeling the TOE
ALC_CMS.1	TOE CM coverage
AGD: CC assurance class guidance documents	
AGD_OPE.1	Operational user guidance
AGD_PRE.1	Preparative procedures
ADV: CC assurance class development	
ADV_FSP.1	Basic functional specification
ATE: CC assurance class tests	
ATE_IND.1	Independent testing - conformance
AVA: CC assurance class vulnerability assessment	
AVA_VAN.1	Vulnerability analysis

Table 20: Overview of SARs

6.2.2 ASE: Security target evaluation

6.2.2.1 ASE_CCL.1 Conformance claims

Dependencies: ASE_INT.1
ASE_ECD.1
ASE_REQ.2

Operations: None.

1020 **ASE_CCL.1.1D** The developer shall provide a conformance claim.

ASE_CCL.1.2D The developer shall provide a conformance claim rationale.

ASE_CCL.1.1C The conformance claim shall contain a CC conformance claim that identifies the version of the CC to which the ST and the TOE claim conformance.

ASE_CCL.1.2C The CC conformance claim shall describe the conformance of the ST to CC Part 2 as either CC Part 2 conformant or CC Part 2 extended.

ASE_CCL.1.3C The CC conformance claim shall describe the conformance of the ST to CC Part 3 as either CC Part 3 conformant or CC Part 3 extended.

ASE_CCL.1.4C The CC conformance claim shall be consistent with the extended components definition.

ASE_CCL.1.5C The conformance claim shall identify all PPs and security requirement packages to which the ST claims conformance.

ASE_CCL.1.6C The conformance claim shall describe any conformance of the ST to a package as either package-conformant or package-augmented.

ASE_CCL.1.7C The conformance claim rationale shall demonstrate that the TOE type is consistent with the TOE type in the PPs for which conformance is being claimed.

ASE_CCL.1.8C The conformance claim rationale shall demonstrate that the statement of the security problem definition is consistent with the statement of the security problem definition in the PPs for which conformance is being claimed.

ASE_CCL.1.9C The conformance claim rationale shall demonstrate that the statement of security objectives is consistent with the statement of security objectives in the PPs for which conformance is being claimed.

ASE_CCL.1.10C The conformance claim rationale shall demonstrate that the statement of security requirements is consistent with the statement of security requirements in the PPs for which conformance is being claimed.

ASE_CCL.1.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

6.2.2.2 ASE_ECD.1 Extended components definition

Dependencies: None.

Operations: None.

ASE_ECD.1.1D The developer shall provide a statement of security requirements.

ASE_ECD.1.2D The developer shall provide an extended components definition.

ASE_ECD.1.1C The statement of security requirements shall identify all extended security requirements.

ASE_ECD.1.2C The extended components definition shall define an extended component for each extended security requirement.

ASE_ECD.1.3C The extended components definition shall describe how each extended component is related to the existing CC components, families, and classes.

ASE_ECD.1.4C The extended components definition shall use the existing CC components, families, classes, and methodology as a model for presentation.

ASE_ECD.1.5C The extended components shall consist of measurable and objective elements such that conformance or nonconformance to these elements can be demonstrated.

ASE_ECD.1.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

ASE_ECD.1.2E The evaluator *shall confirm* that no extended component can be clearly expressed using existing components.

6.2.2.3 ASE_INT.1 ST Introduction

Dependencies: None.

Operations: Refinements.

1070 **ASE_INT.1.1D** The developer shall provide an ST introduction.

ASE_INT.1.1C The ST introduction shall contain an ST reference, a TOE reference, a TOE overview and a TOE description.

ASE_INT.1.2C The ST reference shall uniquely identify the ST.

ASE_INT.1.3C The TOE reference shall identify the TOE.

ASE_INT.1.4C The TOE overview shall summarise the usage and major security features of the TOE.

ASE_INT.1.5C The TOE overview shall identify the TOE type.

ASE_INT.1.6C The TOE overview shall identify any non-TOE hardware/software/firmware required by the TOE.

1080 **ASE_INT.1.7C** The TOE description shall describe the physical scope of the TOE.

ASE_INT.1.8C The TOE description shall describe the logical scope of the TOE.

ASE_INT.1.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

ASE_INT.1.2E The evaluator *shall confirm* that the TOE reference, the TOE overview, and the TOE description are consistent with each other.

Refinement: **NDPP-33E**
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.

1090 ### 6.2.2.4 ASE_OBJ.2 Security objectives

Dependencies: ASE_SPD.1

Operations: None.

ASE_OBJ.2.1D The developer shall provide a statement of security objectives.

ASE_OBJ.2.2D The developer shall provide a security objectives rationale.

ASE_OBJ.2.1C The statement of security objectives shall describe the security objectives for the TOE and the security objectives for the operational environment.

ASE_OBJ.2.2C The security objectives rationale shall trace each security objective for the TOE back to threats countered by that security objective and OSPs enforced by that security objective.

1100 **ASE_OBJ.2.3C** The security objectives rationale shall trace each security objective for the operational environment back to threats countered by that security objective, OSPs enforced by that security objective, and assumptions upheld by that security objective.

ASE_OBJ.2.4C The security objectives rationale shall demonstrate that the security objectives counter all threats.

ASE_OBJ.2.5C The security objectives rationale shall demonstrate that the security objectives enforce all OSPs.

ASE_OBJ.2.6C The security objectives rationale shall demonstrate that the security objectives for the operational environment uphold all assumptions.

1110 **ASE_OBJ.2.1E** The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

6.2.2.5 ASE_REQ.2 Derived security requirements

Dependencies: ASE_OBJ.2
ASE_ECD.1

Operations: None.

ASE_REQ.2.1D The developer shall provide a statement of security requirements.

ASE_REQ.2.2D The developer shall provide a security requirements rationale.

ASE_REQ.2.1C The statement of security requirements shall describe the SFRs and the SARs.

ASE_REQ.2.2C All subjects, objects, operations, security attributes, external entities and other
1120 terms that are used in the SFRs and the SARs shall be defined.

ASE_REQ.2.3C The statement of security requirements shall identify all operations on the security requirements.

ASE_REQ.2.4C All operations shall be performed correctly.

ASE_REQ.2.5C Each dependency of the security requirements shall either be satisfied, or the security requirements rationale shall justify the dependency not being satisfied.

ASE_REQ.2.6C The security requirements rationale shall trace each SFR back to the security objectives for the TOE.

ASE_REQ.2.7C The security requirements rationale shall demonstrate that the SFRs meet all security objectives for the TOE.

1130 **ASE_REQ.2.8C** The security requirements rationale shall explain why the SARs were chosen.

ASE_REQ.2.9C The statement of security requirements shall be internally consistent.

ASE_REQ.2.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

6.2.2.6 ASE_SPD.1 Security problem definition

Dependencies: None.

Operations: None.

ASE_SPD.1.1D The developer shall provide a security problem definition.

ASE_SPD.1.1C The security problem definition shall describe the threats.

ASE_SPD.1.2C All threats shall be described in terms of a threat agent, an asset, and an adverse
1140 action.

ASE_SPD.1.3C The security problem definition shall describe the OSPs.

ASE_SPD.1.4C The security problem definition shall describe the assumptions about the operational environment of the TOE.

ASE_SPD.1.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

6.2.2.7 ASE_TSS.1 TOE summary specification

Dependencies: ASE_INT.1

ASE_REQ.2

ADV_FSP.1

Operations: Refinements.

ASE_TSS.1.1D The developer shall provide a TOE summary specification.

ASE_TSS.1.1C The TOE summary specification shall describe how the TOE meets each SFR.

ASE_TSS.1.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

ASE_TSS.1.2E The evaluator *shall confirm* that the TOE summary specification is consistent with the TOE overview and the TOE description.

Refinement: **NDPP-07E**

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").

Refinement: **NDPP-12E**

The evaluator shall review the TSS section to determine the version number of the product containing the RBG(s) used in the TOE. The evaluator shall also confirm that the TSS describes the hardware-based noise source from which entropy is gathered, and confirm the location of this noise source. The evaluator will further verify that all of the underlying functions and parameters used in the RBG are listed in the TSS.

Refinement: **NDPP-13E**

The evaluator shall verify that the TSS contains a description of the RBG model, including the method for obtaining entropy input, as well as identifying the entropy source(s) used and how much entropy is produced by each entropy source. The evaluator shall also ensure that the TSS describes known modes of entropy source failure. Finally, the evaluator shall ensure that the TSS contains a description of the RBG outputs in terms of the independence of the output and variance with time and/or environmental conditions.

Refinement: **NDPP-15E**

”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 concains 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 ad a minimum describes how the previous data are zeroized/overwritten, and at what point in the buffer processing this occurs.

Refinement: **NDPP-19E**

The evaluator shall examine the TSS to determine that it details how any plaintext 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. If passwords are not stored in plaintext, the TSS shall describe how the passwords are protected.

Refinement: **NDPP-20E**

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.

Refinement: **NDPP-21E**

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.

Refinement: **NDPP-23E**

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.

Refinement: **NDPP-26E**

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).

Refinement: **NDPP-A-01E**
The evaluator shall examine the TSS to verify that it describes how "confidentiality only" ESP mode is disabled.

Refinement: **NDPP-A-03E**
The evaluator shall examine the TSS to ensure that, in the description of the IPSec protocol supported by the TOE, it states that aggressive mode is not used for IKEv1 Phase 1 exchanges, and that only main mode is used. If this requires configuration of the TOE prior to its operation, the evaluator shall check the operational guidance to ensure that instructions for this configuration are contained within that guidance.

Refinement: **NDPP-A-06E**
The evaluator checks to ensure that the TSS describes how lifetimes for IKEv1 SAs (both Phase 1 and Phase 2) are established.

Refinement: **NDPP-A-09E**
The evaluator checks to ensure that the TSS describes how lifetimes for IKEv1 Phase 2 SAs--with respect to the amount of traffic that is allowed to flow using a given SA--are established. If the value is configurable, then the evaluator verifies that the appropriate instructions for configuring these values are included in the operational guidance.

Refinement: **NDPP-A-12E**
The evaluator shall check to ensure that the DH groups specified in the requirement are listed as being supported in the TSS. If there is more than one DH group supported, the evaluator checks to ensure the TSS describes how a particular DH group is specified/negotiated with a peer.

Refinement: **NDPP-A-14E**
The evaluator shall check that the TSS contains a description of the IKE peer authentication process used by the TOE, and that this description covers the use of the signature algorithm or algorithms specified in the requirement.

Refinement: **NDPP-A-16E**
The evaluator shall check to ensure that the TSS describes how pre-shared keys are established and used in authentication of IPSec connections. The evaluator shall check that the operational guidance describes how pre-shared keys are to be generated and established for a TOE. The description in the TSS and the operational guidance shall also indicate how preshared key establishment is accomplished for both TOEs that can generate a pre-shared key as well as TOEs that simply use a pre-shared key.

Refinement: **NDPP-A-22E**
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.

Refinement: **NDPP-A-25E**
The evaluator shall examine the TSS to ensure that it specifies that the TOE

rekeys an SSH connection before more than 2^{28} packets have been sent with a given key. If this effect is achieved by configuration of the TOE, then the evaluator shall examine the operational guidance to ensure that it contains instructions on setting the appropriate values.

Refinement: **NDPP-A-27E**

The evaluator shall check to ensure that the TSS specifies the timeout period and the method for dropping a session connection after the number of failed authentication attempts specified in the requirement. If these values are configurable and may be specified by the security administrator, the evaluator shall check the operational guidance to ensure that it contains instructions for configuring these values.

Refinement: **NDPP-A-30E**

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.1.7, and ensure that password-based authentication methods are also allowed.

Refinement: **NDPP-A-32E**

The evaluator shall check that the TSS describes how "large packets" in terms of RFC 4253 are detected and handled.

Refinement: **NDPP-A-34E**

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.

Refinement: **NDPP-A-37E**

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

Refinement: **NDPP-A-38E**

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.

Refinement: **NDPP-A-41E**

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.

Refinement: **NDPP-A-43E**

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.

Refinement: **NDPP-A-01C**
Depending on the specific requirements selected by the ST author from Section C1.1, the ST author should include the appropriate auditable events in the corresponding table in the ST for the requirements selected.

6.2.3 ALC: Life cycle support

6.2.3.1 ALC_CMC.1 Lableing the TOE

Dependencies: ALC_CMS.1

Operations: Refinements.

ALC_CMC1.1D 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.2.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

Refinement: **NDPP-34E**
The "evaluation evidence required by the SARs" in this PP 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.

6.2.3.2 ALC_CMS.1 TOE CM coverage

Dependencies: No dependencies.

Operations: None.

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; 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.

6.2.4 AGD: Guidance documents

6.2.4.1 AGD_OPE.1 Operational user guidance

Dependencies: ADV_FSP.1

Operations: Refinements.

AGD_OPE.1.1D The developer shall provide operational user guidance.

AGD_OPE.1.1C The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.

AGD_OPE.1.2C The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.

AGD_OPE.1.3C The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.

AGD_OPE.1.4C The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_OPE.1.5C The operational user guidance shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AGD_OPE.1.6C The operational user guidance shall, for each user role, describe the security measures to be followed in order to fulfil the security objectives for the operational environment as described in the ST.

AGD_OPE.1.7C The operational user guidance shall be clear and reasonable.

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

Refinement: **NDPP-01E**
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 PP 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.

Refinement: **NDPP-02E**
The evaluator shall also make a determination of the administrative actions that are relevant in the context of this PP. 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 PP. The evaluator shall document the methodology or approach taken while determining which actions in the administrative guide are security relevant with respect to this PP. The evaluator may perform this activity as part of the activities associated with ensuring the AGD_OPE guidance satisfies the requirements.

Refinement: **NDPP-04E**
The evaluator shall examine the administrative guidance to ensure it instructs the administrator how to establish communication with the audit server. The guidance must instruct how this channel is established in a secure manner (e.g., IPSec, TLS). The evaluator checks the administrative guidance to determine what action(s) is taken if the link between the TOE and audit server is broken. This could be due to network connectivity being lost, or the secure protocol link being terminated.

Refinement: **NDPP-05E**

The evaluator shall test the administrative guidance by establishing a link to the audit server. Note that this will need to be done in order to perform the assurance activities prescribed under FAU_GEN.1. The evaluator shall disrupt the communication link (e.g., unplug the network cable, terminate the protocol link, shutdown the audit server) to determine that the action(s) described in the administrative guide appropriately take place.

Refinement: **NDPP-16E**

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 formulation and specification of password composition rules and how to configure these for the TOE; and how to configure the maximum lifetime for a password.

Refinement: **NDPP-01C**

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.

Refinement: **NDPP-02C**

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.

Refinement: **NDPP-29E**

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.

Refinement: **NDPP-A-02E**

The evaluator shall also examine the operational guidance to determine that

it describes any configuration necessary to ensure that "confidentiality only" mode is disabled, and that an advisory is present indicating that tunnel mode is the preferred ESP mode since it protects the entire packet.

Refinement: **NDPP-A-04E**
The evaluator shall examine the TSS to ensure that, in the description of the IPSec protocol supported by the TOE, it states that aggressive mode is not used for IKEv1 Phase 1 exchanges, and that only main mode is used. If this requires configuration of the TOE prior to its operation, the evaluator shall check the operational guidance to ensure that instructions for this configuration are contained within that guidance.

Refinement: **NDPP-A-07E**
The evaluator checks to ensure that the TSS describes how lifetimes for IKEv1 SAs (both Phase 1 and Phase 2) are established. If they are configurable, then the evaluator verifies that the appropriate instructions for configuring these values are included in the operational guidance.

Refinement: **NDPP-A-10E**
The evaluator checks to ensure that the TSS describes how lifetimes for IKEv1 Phase 2 SAs--with respect to the amount of traffic that is allowed to flow using a given SA--are established. If the value is configurable, then the evaluator verifies that the appropriate instructions for configuring these values are included in the operational guidance.

Refinement: **NDPP-A-17E**
The evaluator shall check that the operational guidance describes how pre-shared keys are to be generated and established for a TOE. The description in the TSS and the operational guidance shall also indicate how pre-shared key establishment is accomplished for both TOEs that can generate a pre-shared key as well as TOEs that simply use a pre-shared key.

Refinement: **NDPP-A-20E**
The evaluator shall check the operational guidance to ensure that it describes the generation of pre-shared keys, including guidance on generating strong keys and the allowed character set. The evaluator shall check that this guidance does not limit the pre-shared key in a way that would not satisfy the requirement. It should be noted that while the administrator (in contravention to the operational guidance) can choose a key that does not conform to the requirement, there is no requirement that the TOE check the key to ensure that it meets the rules specified in this component. However, should the administrator choose to create a password that conforms to the rules above (and the operational guidance); the TOE should not prohibit such a choice.

Refinement: **NDPP-A-23E**
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).

Refinement: **NDPP-A-26E**
The evaluator shall examine the TSS to ensure that it specifies that the TOE rekeys an SSH connection before more than 228 packets have been sent with

a given key. If this effect is achieved by configuration of the TOE, then the evaluator shall examine the operational guidance to ensure that it contains instructions on setting the appropriate values.

Refinement: **NDPP-A-28E**

The evaluator shall check to ensure that the TSS specifies the timeout period and the method for dropping a session connection after the number of failed authentication attempts specified in the requirement. If these values are configurable and may be specified by the security administrator, the evaluator shall check the operational guidance to ensure that it contains instructions for configuring these values.

Refinement: **NDPP-A-35E**

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).

Refinement: **NDPP-A-39E**

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).

Refinement: **NDPP-A-40E**

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.

6.2.4.2 AGD_PRE.1 Preparative procedures

Dependencies: No dependencies.

Operations: Refinements.

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.

Refinement: **NDPP-30E**

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.

6.2.5 ADV: Development

6.2.5.1 ADV_FSP.1 Basic functional specification

Dependencies: No dependencies

Operations: None.

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.

6.2.6 ATE: Tests

6.2.6.1 ATE_IND.1 Independent testing – conformance

Dependencies: ADV_FSP.1 Basic functional specification
AGD_OPE.1 Operational user guidnace
AGD_PRE.1 preparative procedures

Operations: Refinements.

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.

Refinement: **NDPP-03E**
The evaluator shall test the TOE's ability to correctly generate audit records by having the TOE generate audit records for the following events: the establishment and termination of channels, detection of a replay attack, and

administrative actions. The evaluator shall test that the establishment and termination of a channel is performed for each of the cryptographic protocols contained in the PP (i.e., IPSec, SSH, TLS, HTTPS). The test demonstrating the establishment and termination of a TLS session can be combined with the test for an HTTPS session. For the replay attack, the evaluator shall test that a replay audit event can be generated when encountered by each of the cryptographic protocols contained in the PP. For administrative actions, the evaluator shall test that each action determined by the evaluator above to be security relevant in the context of this PP 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 to ensure the TOE can detect replay attempts will more than likely be done to demonstrate that requirement FPT_RPL.1 is satisfied. Another example is that 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.

Refinement: **NDPP-06E**

The evaluator shall use the domain parameter generation and key pair generation portions of "The FIPS 186-3 Digital Signature Algorithm Validation System (DSAVS)", "The FIPS 186-3 Elliptic Curve Digital Signature Algorithm Validation System (ECDSA2VS)", and "The RSA Validation System (RSAVS)" 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.

Refinement: **NDPP-08E**

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.

Refinement: **NDPP-09E**

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.

Refinement: **NDPP-10E**

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.

Refinement: **NDPP-11E**

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.

Refinement: **NDPP-14E**

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

Implementations Conforming to FIPS 140-2, Annex C

The reference for the tests contained in this section is The Random Number Generator Validation System (RNGVS) [RNGVS]. The evaluator shall conduct the following two tests. Note that the "expected values" are produced by a reference implementation of the algorithm that is known to be correct. Proof of correctness is left to each Scheme.

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

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

Implementations Conforming to NIST Special Publication 800-90

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

If the RBG has prediction resistance enabled, each trial consists of (1) instantiate drbg, (2) generate the first block of random bits (3) generate a second block of random bits (4) uninstantiate. The evaluator verifies that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 – 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The next two are additional input and entropy input for the first call to generate. The final two are additional input and entropy input for the second call to generate. These values are randomly generated. "generate one block of random bits" means to generate random bits with

number of returned bits equal to the Output Block Length (as defined in NIST SP 800-90).

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

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

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

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

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

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

Refinement: **NDPP-17E**
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 configure the TOE with different password composition rules, as specified in the requirement. The evaluator shall then, for each set of rules, 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 composition rules are enforced. While the evaluator is not required (nor is it feasible) to test all possible composition rules, 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.

Test 2: The evaluator shall ensure that the operational guidance contains instructions on setting the maximum password lifetime. The evaluator shall then configure this lifetime to several values, and ensure that it is enforced for each of those values.

Test 3: The evaluator shall test that a minimum of 4 character changes from previous passwords is enforced. This shall be done for more than one password.

Refinement: **NDPP-18E**
The evaluator shall perform the following test:

Test 1: The evaluator shall attempt to change their password as directed by the operational guidance. While making this attempt, the evaluator shall verify that re-authentication is required.

Refinement: **NDPP-22E**
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.

Refinement: **NDPP-24E**
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.

Refinement: **NDPP-25E**
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.

Refinement: **NDPP-27E**
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.

Refinement: **NDPP-31E**
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 this PP'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 this PP 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.

Refinement: **NDPP-A-05E**
The evaluator shall also perform the following tests:

Test 1: The evaluator shall configure the TOE as indicated in the operational guidance, and attempt to establish a connection using an IKEv1 Phase 1 connection in aggressive mode. This attempt should fail. The evaluator should then show that main mode exchanges are supported.

Test 2: The evaluator shall configure the TOE as indicated in the operational guidance, and attempt to establish a connection using ESP in "confidentiality only" mode. This attempt should fail. The evaluator shall then establish a connection using ESP in confidentiality and integrity mode.

Refinement: **NDPP-A-08E**
The evaluator also performs the following test:

Test 1: The evaluator shall construct a test where a Phase 1 SA is established and attempted to be maintained for more than 24 hours before it is renegotiated. The evaluator shall observe that this SA is closed or renegotiated in 24 hours or less. If such an action requires that the TOE be configured in a specific way, the evaluator shall implement tests demonstrating that the configuration capability of the TOE works as documented in the operational guidance.

Test 2: The evaluator shall perform a test similar to Test 1 for Phase 2 SAs, except that the lifetime will be 8 hours instead of 24.

Refinement: **NDPP-A-11E**
The evaluator also performs the following test:

Test 1: The evaluator shall construct a test where a Phase 2 SA is established and attempted to be maintained while more data than is specified in the above assignment flows over the connection. The evaluator shall observe that this SA is closed or renegotiated before the amount of data specified is exceeded. If such an action requires that the TOE be configured

in a specific way, the evaluator shall implement tests demonstrating that the configuration capability of the TOE works as documented in the operational guidance.

Refinement: **NDPP-A-13E**
The evaluator shall also perform the following test:

Test 1: For each supported DH group, the evaluator shall test to ensure that all IKE protocols can be successfully completed using that particular DH group.

Refinement: **NDPP-A-15E**
The evaluator shall also perform the following test:

Test 1: For each supported signature algorithm, the evaluator shall test that peer authentication using that algorithm can be successfully achieved.

Refinement: **NDPP-A-18E**
The evaluator shall also perform the following test:

Test 1: The evaluator shall generate a pre-shared key and use it, as indicated in the operational guidance, to establish and IPSec connection between two peers. If the TOE supports generation of the pre-shared key, the evaluator shall ensure that establishment of the key is carried out for an instance of the TOE generating the key as well as an instance of the TOE merely taking in and using the key.

Refinement: **NDPP-A-21E**
The evaluator shall also perform the following test; this may be combined with Test 1 for FCS_IPSEC_EXT.1.7:

Test 1: The evaluator shall generate a pre-shared key that is 22 characters long that meets the composition requirements above. The evaluator shall then use this key to successfully establish an IPSec connection. While the evaluator is not required to test that all of the special characters or lengths listed in the requirement are supported, it is required that they justify the subset of those characters chosen for testing, if a subset is indeed used.

Refinement: **NDPP-A-24E**
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).

Refinement: **NDPP-A-29E**
The evaluator shall also perform the following tests:

Test 1: The evaluator shall demonstrate that taking longer than the timeout period to authenticate to the TOE results in a disconnection of the current session and requires that the evaluator initiate a new session to attempt to connect. If the timeout period is configurable, the evaluator shall ensure that the operational guidance is followed to implement at least two different periods in order to ensure that the mechanism works as specified.

Test 2: The evaluator shall demonstrate that performing a number of failed SSH authentication attempts equal to the value specified in the requirement results in a disconnection of the current session and requires that the evaluator initiate a new session to attempt to connect. If this number is configurable, the evaluator shall ensure that the operational guidance is followed to implement at least two different limits (e.g., 3 attempts and 5 attempts) in order to ensure that the mechanism works as specified.

Refinement: **NDPP-A-31E**
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.

Refinement: **NDPP-A-33E**
The evaluator shall also perform the following tests:

Test 1: The evaluator shall demonstrate that if the TOE receives a packet larger than that specified in this component (FCS_SSH_EXT.1.5), that packet is dropped.

Refinement: **NDPP-A-36E**
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.

Refinement: **NDPP-A-42E**
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

Refinement: **NDPP-A-44E**
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.

6.2.7 AVA: Vulnerability assessment

6.2.7.1 AVA_VAN.1 Vulnerability analysis

Dependencies:

ADV_FSP.1 Basic functional specification

AGD_OPE.1 Operational user gudiance
AGD_PRE.1 Preparative procedures

Operations: Refinements.

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.

Refinement: **NDPP-32E**

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 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 this PP. 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.

6.3 Security requirements rationale

According to the two groups SFR and SAR of security requirements, the security requirements rationale is also divided into two sections: Security functional requirements rationale and security assurance requirements rationale.

6.3.1 Security functional requirements rationale

The security functional requirements rationale contains a tracing of SFRs to security objectives of the TOE and a set of justifications that shows that all security objectives for the TOE are effectively addressed by the SFRs.

6.3.1.1 Tracing of SFRs to security objectives of the TOE

The following table shows how the SFRs trace back to security objectives of the TOE.

	O.PROTECTED_COMMUNICATIONS	O.VERIFIABLE_UPDATES	O.SYSTEM_MONITORING	O.DISPLAY_BANNER	O.TOE_ADMINISTRATION	O.RESIDUAL_INFORMATION_CLEARING	O.RESOURCE_AVAILABILITY	O.SESSION_LOCK	O.TSF_SELF_TEST
FAU_GEN.1			X		X				
FAU_GEN.2			X						
FAU_STG_EXT.1	X		X		X				
FAU_STG_EXT.3	X		X						
FCS_CKM.1	X								
FCS_CKM_EXT.4	X								
FCS_COP.1(1)	X								
FCS_COP.1(2)	X	X							
FCS_COP.1(3)	X	X							
FCS_COP.1(4)	X								
FCS_RBG_EXT.1	X								
FCS_COMM_PROT_EXT.1	X								
FCS_IPSEC_EXT.1	X		X						
FDP_RIP.2						X			
FIA_PMG_EXT.1					X				
FIA_UIA_EXT.1					X				
FIA_UAU_EXT.5					X				
FIA_UAU.6					X				
FIA_UAU.7					X				
FMT_MTD.1					X				
FMT_SMF.1					X				
FMT_SMR.1					X				
FPT_ITT.1(1)	X								
FPT_ITT.1(2)	X								
FPT_PTD_EXT.1(1)	X				X				
FPT_PTD_EXT.1(2)	X								
FPT_RPL.1	X								
FPT_STM.1			X		X				
FPT_TUD_EXT.1		X							
FPT_TST_EXT.1									X
FRU_RSA.1							X		
FTA_SSL_EXT.1					X			X	
FTA_SSL.3					X			X	
FTA_TAB.1				X					
FTP_ITC.1(1)	X								
FTP_ITC.1(2)	X								
FTP_TRP.1(1)	X								

	O.PROTECTED_COMMUNICATIONS	O.VERIFIABLE_UPDATES	O.SYSTEM_MONITORING	O.DISPLAY_BANNER	O.TOE_ADMINISTRATION	O.RESIDUAL_INFORMATION_CLEARING	O.RESOURCE_AVAILABILITY	O.SESSION_LOCK	O.TSF_SELF_TEST
FTP_TRP.1(2)	X								

Table 21: Tracing of SFRs to security objectives of the TOE

The inspection of Table 21: Tracing of SFRs to security objectives of the TOE shows that:

- Each SFR traces back to at least one security objective,
- each security objective for the TOE has at least one SFR tracing to it.

6.3.1.2 Justification of SFR tracing

The justification demonstrates that the SFRs address all security objectives of the TOE.

Objective	Rationale
Security Functional Requirements Drawn from Security Requirements for NDPP	
O.PROTECTED_COMMUNICATIONS	The SFRs, FAU_STG_EXT.1, FAU_STG_EXT.3, FCS_CKM.1,FCS_CKM_EXT.4, FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3), FCS_COP.1(4), FCS_RBG_EXT.1, FCS_COMM_PROT_EXT.1 FCS_IPSEC_EXT.1, FPT_PTD_EXT.1(2), FPT_RPL.1, FTP_ITC.1(1), FTP_ITC.1(2), FTP_TRP.1(1), FTP_TRP.1(2) meet this objective by ensuring the communications between the TOE and endpoints are secure by implementing the encryption protocols as defined in the SFRs and as specified by the RFCs.
O.VERIFIABLE_UPDATES	The SFRs, FPT_TUD_EXT.1, FCS_COP.1(2), FCS_COP.1(3) meet this objective by ensuring the update was downloaded via secure communications, is from a trusted source, and the update can be verified by cryptographic mechanisms prior to installation.
O.SYSTEM_MONITORING	The SFRs, FAU_GEN.1, FAU_GEN.2, FAU_STG_EXT.1, FAU_STG_EXT.3, FCS_IPSEC_EXT.1, FPT_STM.1 meet this objective by auditing actions on the TOE. The audit records identify the user associated with the action/event, whether the action/event was successful or failed, the type of action/event, and the date/time the action/event occurred. The audit logs are transmitted securely to a remote syslog server. If connectivity to the remote syslog server is

Objective	Rationale
	lost, the TOE will block new permit actions. The TOE will provide the authorized administrators the capability to review Audit data. The TOE does not have an interface to modify audit records, though there is an interface available for the authorized administrator to delete audit data stored locally on the TOE as provided by FAU_STG_EXT.1.
O.DISPLAY_BANNER	The SFR, FTA_TAB.1 meets this objective by displaying a advisory notice and consent warning message regarding unauthorized use of the TOE.
O.TOE_ADMINISTRATION	The SFRs, FIA_UIA_EXT.1, FIA_PMG_EXT.1, FIA_UAU_EXT.5, FIA_UAU.6, FIA_UAU.7, FMT_MTD.1, FMT_SMF.1, FMT_SFR.1, FPT_PTD_EXT.1(1), FTA_SSL_EXT.1, FTA_SSL.3 meet this objective by ensuring the TOE supports a password-based authentication mechanism with password complexity enforcement such as, strong passwords, password life-time constraints, providing current password when changing the password, obscured password feedback when logging in, and passwords are not stored in plaintext. TOE provides the management and configuration features to securely manage the TOE and that those functions are restricted to the authorized administrator, and the implementation of session termination after an administrative configurable inactivity time period whereas the user must be re-authenticated. The TOE must also protect itself against attempts by unauthorized users to bypass, deactivate, or tamper with TOE security functions. The TOE does not have an interface to modify audit records, though there is an interface available for the authorized administrator to delete audit data stored locally on the TOE as provided by FAU_STG_EXT.1, FAU_GEN.1, and FPT_STM.1.
O.RESIDUAL_INFORMATION_CLEARING	The SFR, FDP_RIP.2 meets this objective by ensuring no left over user data from the previous transmission is included in the network traffic.
O.RESOURCE_AVAILABILITY	The SFR, FRU_RSA.1 meets this objective by limiting the number of amount of exhaustible resources, such the number of concurrent administrative sessions.
O.SESSION_LOCK	The SFRs, FTA_SSL_EXT.1, FTA_SSL.3 meet this objective by terminating a session due to meeting/exceeding the inactivity time limit.
O.TSF_SELF_TEST	The SFR, FPT_TST_EXT.1 meets this objective by performing self-test to ensure the TOE is operating correctly and all functions are available and enforced.

Table 22: Security objectives rationale

6.3.1.3 Fulfillment of SFR dependencies

The following table summarizes the existing dependencies between security requirements.

SFR	Dependency	Rationale
FAU_GEN.1	FPT_STM.1	Met by FPT_STM.1
FAU_GEN.2	FAU_GEN.1 FIA_UID.1	Met by FAU_GEN.1 Met by FIA_UIA_EXT.1
FAU_STG_EXT.1	FAU_GEN.1 FCS_IPSEC_EXT.1	Met by FAU_GEN.1 Met by FCS_IPSEC_EXT.1
FAU_STG_EXT.3	none	none
FCS_CKM.1	FCS_CKM.2 or FCS_COP.1 FCS_CKM.4	Met by FCS_COP.1(2), (3), and (4) Met by FCS_CKM_EXT.4
FCS_CKM_EXT.4	FCS_CKM.1	Met by FCS_CKM.1
FCS_COP.1(1)	FDP_ITC.1 or 2 or FCS_CKM.1 FCS_CKM.4	Met by FCS_CKM.1 Met by FCS_CKM_EXT.4
FCS_COP.1(2)	FDP_ITC.1 or 2 or FCS_CKM.1 FCS_CKM.4	Met by FCS_CKM.1 Met by FCS_CKM_EXT.4
FCS_COP.1(3)	FDP_ITC.1 or 2 or FCS_CKM.1 FCS_CKM.4	Met by FCS_CKM.1 Met by FCS_CKM_EXT.4
FCS_COP.1(4)	FDP_ITC.1 or 2 or FCS_CKM.1 FCS_CKM.4	Met by FCS_CKM.1 Met by FCS_CKM_EXT.4
FCS_RBG_EXT.1	No dependencies	N/A
FCS_COMM_PROT_EXT.1	FCS_IPSEC_EXT.1 or FCS_SSH_EXT.1	Met by FCS_IPSEC_EXT.1
FCS_IPSEC_EXT.1	No dependencies	N/A
FDP_RIP.2	No dependencies	N/A
FIA_PMG_EXT.1	No dependencies	N/A
FIA_UIA_EXT.1	No dependencies	N/A
FIA_UAU_EXT.5	No dependencies	N/A
FIA_UAU.6	No dependencies	N/A
FIA_UAU.7	FIA_UAU.1	Met by FIA_UIA_EXT.1
FMT_MTD.1	FMT_SMF.1 FMT_SMR.1	Met by FMT_SMF.1 Met by FMT_SMR.1
FMT_SMF.1	No dependencies	N/A
FMT_SMR.1	FIA_UID.1	Met by FIA_UIA_EXT.1
FPT_PTD_EXT.1(1)	No dependencies	N/A

SFR	Dependency	Rationale
FPT_PTD_EXT.1(2)	FCS_CKM.1	Met by FCS_CKM.1
FPT_RPL.1	No dependencies	N/A
FPT_STM.1	No dependencies	N/A
FPT_TUD_EXT.1	No dependencies	N/A
FPT_TST_EXT.1	No dependencies	N/A
FRU_RSA.1	No dependencies	N/A
FTA_SSL_EXT.1	No dependencies	N/A
FTA_SSL.3	No dependencies	N/A
FTA_TAB.1	No dependencies	N/A
FTP_ITC.1(1)	No dependencies	N/A
FTP_ITC.1(2)	No dependencies	N/A
FTP_TRP.1(1)	No dependencies	N/A
FTP_TRP.1(2)	No dependencies	N/A

Table 23: Fulfillment of SFR dependencies

The inspection of Table 23: Fulfillment of SFR dependencies shows that:

- All dependencies between security functional requirements and other security requirements are satisfied, except as noted below.

6.3.1.4 Justification of missing SFR dependencies

The dependencies that are met by FCS_CKM.4 are statisfied by the NDPP extended SFR FCS_CKM_EXT.4. The NDPP version of the SFR identifies the keys that must be zerioed when no longer needed and the method in which they must be destroyed. This satisfies the intent of the CC SFR, therefore the dependency is satisfied.

The dependencies that are met by FIA_UID.1 or FIA_UAU.1 are statisfied by the NDPP extended SFR FIA_UIA_EXT.1. The NDPP version of the SFR allows for the list of services that are provided prior to the user being identified and authenticated. This satisfies the intent of the CC SFR and includes identification and authentication, thus the dependency is satisfied.

6.3.2 Security assurance requirements rationale

The security assurance requirements rationale explains why the chosen set of SARs in this ST was deemed appropriate.

6.3.2.1 Set of chosen SARs

The chosen set of SARs for this ST is given in Table 20: Overview of SARs above.

6.3.2.2 Justification of chosen set of SARs

The chosen set of SARs forms the evaluation assurance level as specified in the NDPP.

The package NDPP is already sound and well-defined, and does not introduce new dependencies. Therefore, the justification demonstrates that the chosen set of SARs is reasonable.

6.3.2.3 Fulfillment of SAR dependencies

The following table summarizes the existing dependencies between security requirements.

SAR	Dependencies	Fulfilled by
ASE_CCL.1	ASE_INT.1 ASE_ECD.1 ASE_REQ.2	ASE
ASE_ECD.1	No dependencies	Not necessary
ASE_INT.1	No dependencies	Not necessary
ASE_OBJ.2	ASE_SPD.1	ASE
ASE_REQ.2	ASE_OBJ.2 ASE_ECD.1	ASE
ASE_SPD.1	No dependencies	Not necessary
ASE_TSS.1	ASE_INT.1 ASE_REQ.2 ADV_FSP.1	ASE, ADV
ALC_CMC.1	ALC_CMS.1	ALC
ALC_CMS.1	No dependencies	Not necessary
AGD_OPE.1	ADV_FSP.1	ADV
AGD_PRE.1	No dependencies	Not necessary
ADV_FSP.1	No dependencies	Not necessaryADV
ATE_IND.1	ADV_FSP.1 AGD_OPE.1 AGD_PRE.1	ADV, AGD
AVA_VAN.1	ADV_FSP.1 AGD_OPE.1 AGD_PRE.1	ADV, AGD

Table 24: Fulfillment of SAR dependencies

The inspection of Table 24: Fulfillment of SAR dependencies shows that:

- All dependencies between security assurance requirements and other security requirements are satisfied.

2010

6.3.2.4 Justification of missing SAR dependencies

None at this time.

7 TOE summary specification

The TOE summary specification describes the TOE security functions and how the TOE meets the security functional requirements. The TOE summary specification provides the general technical mechanisms that the TOE uses for this purpose. The description is divided into conceptual groups.

Note that this TOE summary specification contains mostly non-proprietary rationale for each security functional requirement.

The TOE implements the following security functions that together satisfy the SFRs claimed in chapter 6 of this ST:

- SF01: Security Audit,
- SF02: Cryptographic Support,
- SF03: User Data Protection,
- SF04: Identification and Authentication,
- SF05: Security Management,
- SF06: Protection of the TSF,
- SF07: Resource Utilization,
- SF08: TOE Access, and
- SF09: Trusted Path/channels.

7.1 SF01: Security audit

The TOE generates audit records for events that take place on the TOE and activates procedures upon detection of potential security violations.

7.1.1 Security function description

FAU_GEN.1

The TOE generates an audit record whenever an audited event occurs. The types of events that cause audit records to be generated include events related to the enforcement of information flow policies, identification and authentication related events, and administrative events (the specific events and the contents of each audit record are listed in the table within the FAU_GEN.1 SFR, “Auditable Events Table”). Each of the events is specified in the audit record is in enough detail to identify the user for which the event is associated (e.g. user identity, MAC address, IP address), when the event occurred, where the event occurred, the outcome of the event, and the type of event that occurred. Additionally, the startup and shutdown of the audit functionality is audited.

The audit trail consist of the individual audit records; one audit record for each event that occurred. The audit record can contain up to 80 characters and a percent sign (%), which follows the time-stamp information. As noted above, the information includes [at least] all of the required information. Additional information can be configured and included if desired. Refer to the Guidance documentation for configuration syntax and information.

The logging buffer size can be configured from a range of 4096 (default) to 2147483647 bytes. It is noted, not make the buffer size too large because the switch could run out of memory for other tasks. Use the show memory privileged EXEC command to view the free processor memory on the switch. However, this value is the maximum available, and the buffer size should not be set to this amount. Refer to the Guidance documentation for configuration syntax and information.

The administrator can also configure a ‘configuration logger’ to keep track of configuration changes made with the command-line interface (CLI). The administrator can configure the size of the configuration log from 1 to 1000 entries (the default is 100). Refer to the Guidance documentation for configuration syntax and information.

The log buffer is circular, so newer messages overwrite older messages after the buffer is full. Administrators are instructed to monitor the log buffer using the show logging privileged EXEC command to view the audit records. The first message displayed is the oldest message in the buffer. There are other associated commands to clear the buffer, to set the logging level, etc; all of which are described in the Guidance documents and IOS CLI.

The logs can be saved to flash memory so records are not lost in case of failures or restarts. Refer to

the Guidance documentation for configuration syntax and information.

The administrator can set the level of the audit records to be displayed on the console or sent to the syslog server. For instance all emergency, alerts, critical, errors, and warning message can be sent to the console alerting the administrator that some action needs to be taken as these types of messages mean that the functionality of the switch is affected. All notifications and information type message can be sent to the syslog server, where as message is only for information; switch functionality is not affected. Note that audit records are transmitted in the clear to the syslog server, though it is stated the syslog server attached to the internal (trusted) network.

The FIPS crypto tests, the messages are displayed on the console. Once the box is up and operational and the crypto self test command is entered, then the messages would be displayed on the console and will also be logged.

Auditable Event	Rationale
All use of the user identification mechanism.	Events will be generated for attempted identification/ authentication, and the username attempting to authenticate will be included in the log record.
Any use of the authentication mechanism.	Events will be generated for attempted identification/ authentication, and the username attempting to authenticate will be included in the log record, along with the origin or source of the attempt.
Management functions	The use of the security management functions is logged; modifications of the behavior of the functions in the TSF and modifications of default settings.
Detection of replay attacks	Attempts of replaying data previously transmitted and terminated at the TOE are logged, along with the origin or source of the attempt.
Changes to the time.	Changes to the time are logged.
Updates	An audit record will be generated on the initiation of updates (software/firmware)
Failure to establish and/or establishment/failure of an IPsec session	Attempts to establish an IPsec session or the failure of an established IPsec is logged.
Resources quotas are exceeded	If the threshold for the number of concurrent administrative sessions is exceeded, and audit record is generated
Locking and unlocking interactive sessions	Any attempt to unlock an inactive sessions is logged, as is an inactive session when it exceeds the time limit of inactivity
Indication that TSF self-test was completed.	During bootup, if the self test succeeds a login prompt is displayed. If the self-test fails, the failure is logged.
Trusted channels	The initiation, termination, and failure related to trusted channel sessions with peer/neighbor routers and or the remote administration console

FAU_GEN.2

The TOE shall ensure that each auditable event is associated with the user that triggered the event and as a result they are traceable to a specific user. For example a human user, user identity or related session ID would be included in the audit record. For an IT entity or device, the IP address, MAC address, host name, or other configured identification is presented. Refer to the Guidance documentation for configuration syntax and information.

FAU_STG_EXT.1 and FAU_STG_EXT.3

The TOE is configured to export syslog records to a specified, external syslog server. The TOE protects communications with an external syslog server via IPSec. If the IPSec connection fails, the TOE will store audit records on the TOE when it discovers it can no longer communicate with its configured syslog server.

7.1.2 Security function summary

The security function is designed to satisfy the following security functional requirements.

FAU_GEN.1

FAU_GEN.2

FAU_STG_EXT.1

FAU_STG_EXT.3

7.2 SF02: Cryptographic support

The TOE implements cryptographic operations, which provide secure remote access.

7.2.1 Security function description

FCS_CKM.1

The TOE implements a random number generator for RSA key establishment schemes (conformant to NIST SP 800-56B). The TOE is also compliant to ANSI X9.80 (3 January 2000), “Prime Number Generation, Primality Testing, and Primality Certificates” using random integers with deterministic tests. Furthermore, the TOE does not implement elliptic-curve-based key establishment schemes.

FCS_COP.1(1)

The TOE provides symmetric encryption and decryption capabilities using AES in CBC and GCM mode (128, 256 bits) as described in NIST SP 800-38A and NIST SP 800-38D.

FCS_COP.1(2)

The TOE will provide cryptographic signature services using RSA with key size of 2048 and greater as specified in FIPS PUB 186-3, “Digital Signature Standard”.

FCS_COP.1(3)

The TOE provides cryptographic hashing services using SHA-1, SHA-256, SHA-384, and SHA-512 as specified in FIPS Pub 180-3 “Secure Hash Standard.”.

FCS_COP.1(4)

The TOE uses HMAC-SHA1. HMAC-SHA-256, and HMAC-SHA-512 message authentication as part of the RADIUS Key Wrap functionality as specified in FIPS Pub 198-1 “The Keyed-Hash Message Authentication Code” and FIPS PUB 180-3, “Secure Hash Standard”.

2080

FCS_CKM_EXT.4

The TOE meets all requirements specified in FIPS 140-2 for destruction of keys and Critical Security Parameters (CSPs) in that none of the symmetric keys, pre-shared keys, or private keys are stored in plaintext form.

The cryptographic module securely administers both cryptographic keys and other critical security parameters such as passwords. The tamper evidence seals provide physical protection for all keys. All keys are also protected by the password-protection required by the privileged administrator role login, and can be zeroized by the privileged administrator. All zeroization consists of overwriting the memory that stored the key. Keys are exchanged and entered electronically. Persistent keys are entered by the
2090 privileged administrator via the console port CLI, transient keys are generated or established and stored in DRAM. If present, a VSS link can export all DRAM and NVRAM keys to another switch over a secure connection for high availability purposes.

Name	Description	Zeroization
Diffie-Hellman Shared Secret	The value is zeroized after it has been given back to the consuming operation. The value is overwritten by 0’s.	Automatically after completion of DH exchange. Overwritten with: 0x00
Diffie Hellman private exponent	The function returns the value to the RP and then calls the function to perform the zeroization of the generated key pair (p_dh_kepair) and then calls the standard Linux free (without the poisoning). These values are automatically zeroized after generation and once the value has been provided back to the actual consumer.	Zeroized upon completion of DH exchange. Overwritten with: 0x00
Skeyid	The function calls the operation ike_free_ike_sa_chunk, which performs the zeroization of the IKE structure. This structure contains all of the SA items, including the skeyid, skeyid_d, IKE Session Encryption Key and IKE Session Authentication Key. All values overwritten by 0’s.	Automatically after IKE session terminated. Overwritten with: 0x00
skeyid_d	The function calls the operation ike_free_ike_sa_chunk, which performs the zeroization of the IKE structure. This structure contains all of the SA items, including the skeyid, skeyid_d, IKE Session Encryption Key and IKE Session Authentication Key. All values overwritten by 0’s.	Automatically after IKE session terminated. Overwritten with: 0x00
IKE session encrypt key	The function calls the operation ike_free_ike_sa_chunk, which performs the zeroization of the IKE structure. This structure contains all of the SA items, including the skeyid, skeyid_d, IKE Session Encryption Key and IKE Session	Automatically after IKE session terminated. Overwritten with: 0x00

Name	Description	Zeroization
	Authentication Key. All values overwritten by 0's.	
IKE session authentication key	The function calls the operation ike_free_ike_sa_chunk, which performs the zeroization of the IKE structure. This structure contains all of the SA items, including the skeyid, skeyid_d, IKE Session Encryption Key and IKE Session Authentication Key. All values overwritten by 0's.	Automatically after IKE session terminated. Overwritten with: 0x00
ISAKMP preshared	The function calls the free operation with the poisoning mechanism that overwrites the value with 0x0d.	Zeroized using the following command: **# no crypto isakmp key** Overwritten with: 0x0d
IKE RSA Private Key	The operation uses the free operation with the poisoning mechanism that overwrites the value with 0x0d. (This function is used by the module when zeroizing bad key pairs from RSA Key generations.)	Zeroized using the following command: **# crypto key zeroize rsa** Overwritten with: 0x0d
IPSec encryption key	The function zeroizes an _ike_flow structure that includes the encryption and authentication keys. The entire object is overwritten by 0's using memset.	Automatically when IPSec session terminated. Overwritten with: 0x00
IPSec authentication key	The function zeroizes an _ike_flow structure that includes the encryption and authentication keys. The entire object is overwritten by 0's using memset.	Automatically when IPSec session terminated. Overwritten with: 0x00
RADIUS secret	The function calls aaa_free_secret, which uses the poisoned free operation to zeroize the memory from the secret structure by overwriting the space with 0x0d and releasing the memory.	Zeroized using the following command: **# no radius-server key** Overwritten with: 0x0d
TACACS+ secret	The function calls aaa_free_secret, which uses the poisoned free operation to zeroize the memory from the secret structure by overwriting the space with 0x0d and releasing the memory.	Zeroized using the following command: **# no tacacs-server key** Overwritten with: 0x0d

FCS_RBG_EXT.1

The TOE implements a NIST-approved AES-CTR Deterministic Random Bit Generator (DRBG), as specified in SP 800-90.

The entropy source used in the Catalyst 6500 series switches is a chip using analog ring oscillator based noise sources. The solution is available in the 15.1(1)SY1 or later FIPS/CC approved releases of the IOS images relating to the platforms mentioned above. All RNG entropy source samplings are continuously health tested by the NIST DRBG as per SP 900-90A before using them as a seed. Any initialization or system errors during bring-up or processing of this system causes a reboot as necessary.

FCS_COMM_PROT_EXT.1

The TOE implements IPSec used to protect communications for remote administration. IPSec is also used to protect communications with external servers (e.g., syslog server).

FCS_IPSEC_EXT.1

The TOE implements IPSec to provide authentication and encryption services to prevent unauthorized viewing or modification of data as it travels over the external network. The TOE implementation of the IPSec standard (in accordance with the RFCs noted in the SFR) uses the Encapsulating Security Payload (ESP) protocol to provide authentication, encryption and anti-replay services. IPSec Internet Key Exchange, also called ISAKMP, is the negotiation protocol that lets two peers agree on how to build an IPSec SA. IKE separates negotiation into two phases: phase 1 and phase 2. Phase 1 creates the first tunnel, which protects later ISAKMP negotiation messages. The key negotiated in phase 1 enables IKE peers to communicate securely in phase 2. During Phase 2 IKE establishes the IPSec SA. IKE maintains a trusted channel, referred to as a Security Association (SA), between IPSec peers that is also used to manage IPSec connections, including:

- The negotiation of mutually acceptable IPSec options between peers,
- The establishment of additional Security Associations to protect packets flows using ESP, and
- The agreement of secure bulk data encryption AES (128 and 256 bit) keys for use with ESP.

After the two peers agree upon a policy, the security parameters of the policy are identified by an SA established at each peer, and these IKE SAs apply to all subsequent IKE traffic during the negotiation.

The TOE supports both IKEv1 and IKEv2 session establishment. As part of this support, the TOE can be configured to not support aggressive mode for IKEv1 exchanges and to only use main mode using the ‘crypto isakmp aggressive-mode disable’ command.

The TOE can be configured to not allow “confidentiality only” ESP mode by ensuring the IKE Policies configured include ESP-encryption.

The TOE supports configuration lifetimes of both Phase 1 SAs (24 hours) and Phase 2 SAs (8 hours) using the following command, lifetime. The SAs lifetimes can also be configured to limit the amount of traffic and to only implement DH group 14 (2048-bit MODP).

Other configuration options include rDSA algorithm for implementing peer authentication and the pre-shared keys for authenticating IPSec connections can be 22 characters and be composed of any combination of upper and lower case letters, numbers, and special characters.

7.2.2 Security function summary

The security function is designed to satisfy the following security functional requirements.

FCS_CKM.1

FCS_CKM_EXT.4

FCS_COMM_PROT_EXT.1

FCS_COP.1(1)

FCS_COP.1(2)

FCS_COP.1(3)

FCS_COP.1(4)

FCS_IPSEC_EXT.1

FCS_RBG_EXT.1

7.3 SF03: User data protection

The TOE provides various logical architecture structures to assist in controlling information flow through the internal and external networks.

7.3.1 Security function description

The TOE generates/implements/provides …

FDP_RIP.2

The TOE ensures that packets transmitted from the TOE do not contain residual information from previous packets. Packets that are not the required length use zeros for padding. Residual data is never transmitted from the TOE. Once packet handling is completed its content is overwritten before memory buffer which previously contained the packet is reused. This applies to both data plane traffic and administrative session traffic.

7.3.2 Security function summary

The security function is designed to satisfy the following security functional requirements.

FDP_RIP.2

7.4 SF04: Identification and authentication

The TOE provides several identification and authentication mechanisms to allow external entities to interact with the appliance.

7.4.1 Security function description

FIA_PMG_EXT.1

The TOE supports the local definition of users with corresponding passwords. The passwords can be composed of any combination of upper and lower case letters, numbers, and special characters (that include: “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, and “)”. Minimum password length is settable by the Authorized Administrator, and support passwords of 8 characters or greater. Password composition rules specifying the types and number of required characters that comprise the password are settable by the Authorized Administrator. Passwords have a maximum lifetime, configurable by the Authorized Administrator. New passwords must contain a minimum of 4 character changes from the previous password.

FIA_UAU_EXT.5

The TOE can be configured to require local authentication and/or remote authentication via a RADIUS or TACACS+ server as defined in the authentication policy for interactive (human) users. Neighbor routers are authenticated only to passwords stored locally. The policy for interactive (human) users (Administrators) can be authenticated to the local user database, or have redirection to a remote authentication server. Interfaces can be configured to try one or more remote authentication servers, and then fail back to the local user database if the remote authentication servers are inaccessible.

If the interactive (human) users (Administrators) password is expired, the user is required to create a new password after correctly entering the expired password.

FIA_UAU.6

Users changing their passwords are first prompted to enter their old password. Users are also required to their password when re-establishing a remote session due to session termination of inactivity.

The TOE does not provide the capability for an administrator (level 1) to change their own password. However the administrator (level 1) can change their password when required by the TOE (e.g. when expired). At which time the administrator is required to enter their current password before entering a new password. System administrators (level 15) can change any user's password, including their own as required for TOE management, though must be in privilege EXEC mode to perform the function. When the System Administrator (level 15) attempts to change their own password, the TOE will enforce the password expiration policy at which time the System Administrator (level 15) will be required to enter their current password prior to entering a new password. See the Cisco Catalyst 6500-E Series Switches Common Criteria Operational User Guidance and Preparative Procedures for details and configuration settings.

FIA_UAU.7

When a user enters their password at the local console, the ASA displays only '*' characters so that the user password is obscured. For remote session authentication, the TOE does not echo any characters as they are entered.

FIA_UIA_EXT.1

The TOE requires all users to be successfully identified and authenticated before allowing any TSF mediated actions to be performed. Administrative access to the TOE is facilitated through the TOE's CLI. The TOE mediates all administrative actions through the CLI. Once a potential administrative user attempts to access the CLI of the TOE through either a directly connected console or remotely through an IPSec encrypted connection, the TOE prompts the user for a user name and password. Only after the administrative user presents the correct authentication credentials will access to the TOE administrative functionality be granted. No access is allowed to the administrative functionality of the TOE until an administrator is successfully identified and authenticated.

For neighbor routers, which do not have access to the CLI, the neighbor router must present the correct hashed password prior to exchanging routing table updates with the TOE. The TOE authenticates the neighbor router using its supplied password hash, and the source IP address from the IP packet header. The supported routing protocols (BGPv4, EIGRP, RIPv2, OSPFv2, and HSRP) use MD5 hashes to authenticate communications as specified in FCS_COP.1(4).1. For additional security, router protocol traffic can also be isolated to separate VLANs.

7.4.2 Security function summary

The security function is designed to satisfy the following security functional requirements.

FIA_PMG_EXT.1

FIA_UAU_EXT.5

FIA_UAU.6

FIA_UAU.7

FIA_UIA_EXT.1

7.5 SF05: Security management

The TOE implements a series of security management functions.

7.5.1 Security function description

FMT_MTD.1

The TOE provides the ability for authorized administrators to access TOE data, such as audit data, configuration data, security attributes, information flow rules, routing tables, and session thresholds. Each of the predefined and administratively configured privilege level has delete set of permissions that will grant them access to the TOE data, though with some privilege levels, the access is limited. The TOE performs role-based authorization, using TOE platform authorization mechanisms, to grant access to the semi-privileged and privileged levels. For the purposes of this evaluation, the privileged level is equivalent to full administrative access to the CLI, which is the default access for IOS privilege level 15; and the semi-privileged level equates to any privilege level that has a subset of the privileges assigned to level 15. Privilege levels 0 and 1 are defined by default and are customizable, while levels 2-14 are undefined by default and are also customizable. The term “authorized administrator” is used in this ST to refer to any user which has been assigned to a privilege level that is permitted to perform the relevant action; therefore has the appropriate privileges to perform the requested functions. Therefore, semi-privileged administrators with only a subset of privileges can also modify TOE data based if granted the privilege.

FMT_SMF.1

The TOE provides all the capabilities necessary to securely manage the TOE. The administrative user can connect to the TOE using the CLI to perform these functions via IPSec tunnel, a terminal server, or at the local console. Rrefer to the Guidance documentation for configuration syntax, commands, and information related to each of these functions.

The management functionality provided by the TOE includes the following administrative functions:

- Ability to manage the cryptographic functionality - allows the authorized administrator the ability to identify and configure the algorithms used to provide protection of the data, configuration of routing protocols, and if used the configuration of remote authentication
- Ability to manage the audit logs and functions - allows the authorized administrator to configure the audit logs, view the audit logs, and to clear the audit logs
- Ability to manage security attributes belonging to individual users - allows the authorized administrator to create, modify, and delete other administrative users
- Ability to manage the default values of the security attributes - allows the authorized administrator to specify the attributes that are used control access and/or manage users
- Ability to manage the warning banner message and content – allows the authorized administrator the ability to define warning banner that is displayed prior to establishing a session (note this applies to the interactive (human) users; e.g. administrative users
- Ability to manage the time limits of session inactivity – allows the authorized administrator the ability to set and modify the inactivity time threshold.
- Ability to update the TOE and verify the updates are valid.

~~FMT_SMR.1~~

The TOE switch platform maintains administrative privilege level and non-administrative access. Non-administrative access is granted to authenticated neighbor routers for the ability to receive updated routing tables per the information flow rules. There is no other access or functions associated with non-administrative access. The administrative privilege levels include:

- Administrators are assigned to privilege levels 0 and 1. Privilege levels 0 and 1 are defined by default and are customizable. These levels have a very limited scope and access to CLI commands that include basic functions such as login, show running system information, turn on/off privileged commands, logout.
- Semi-privileged administrators equate to any privilege level that has a subset of the privileges assigned to level 15; levels 2-14. These levels are undefined by default and are customizable. The custom level privileges are explained in the example below.
- Privileged administrators are equivalent to full administrative access to the CLI, which is the default access for IOS privilege level 15.

Note, the levels are not hierarchical.

For levels, level 0 is the most restrictive and 15 is the least restrictive.

For level 0, there are five commands associated with privilege level 0: disable, enable, exit, help, and logout. However, the level could be configured to allow a user to have access to the ‘show’ command.

Level 1 is normal EXEC-mode user privileges

Following is an example of how privileges are set and rules in setting privilege levels and assigning users to those privilege levels:

When setting the privilege level for a command with multiple words (commands), the commands starting with the first word will also have the specified access level. For example, if the **show ip route** command is set to level 15, the **show** commands and **show ip** commands are automatically set to privilege level 15—unless they are individually set to different levels. This is necessary because a user cannot execute, for example, the **show ip** command unless the user also has access to **show** commands.

To change the privilege level of a group of commands, the **all** keyword is used. When a group of commands is set to a privilege level using the **all** keyword, all commands which match the beginning string are enabled for that level, and all commands which are available in submodes of that command are enabled for that level. For example, if the **show ip** keywords is set to level 5, **show** and **ip** will be changed to level 5 and all the options that follow the **show ip** string (such as **show ip accounting**, **show ip aliases**, **show ip bgp**, and so on) will be available at privilege level 5.

The **privilege** command is used to move commands from one privilege level to another in order to create the additional levels of administration. The default configuration permits two types of users to access the CLI. The first type of user is a person who is only allowed to access user EXEC mode. The second type of user is a person who is allowed access to privileged EXEC mode. A user who is only allowed to access user EXEC mode is not allowed to view or change the configuration of the networking device, or to make any changes to the operational status of the networking device. On the other hand, a user who is allowed access to privileged EXEC mode can make any change to a networking device that is allowed by the CLI.

Following is an example for setting the privilege levels for staff that are usually not allowed to run all of the commands available in privileged EXEC mode (privilege level 15) on a networking device. They are prevented from running commands that they are not authorized for by not being granted access to the password assigned to **privileged EXEC** mode or to other levels that have been configured on the networking device.

The steps and commands show setting privilege level 7 with access to two commands, clear counters and reload.

Step 1 **enable** password
Enters privileged EXEC mode. Enter the password when prompted.
Router> **enable**
Step 2 **configure terminal**
Enters global configuration mode.
Router# **configure terminal**
Step 3 **enable secret level** *level password*
Configures a new enable secret password for privilege level 7.
Router(config)# **enable secret level** *7 Zy72sKj*
Step 4 **privilege exec level** *level command-string*
Changes the privilege level of the clear counters command from privilege level 15 to privilege level 7.
Router(config)# **privilege exec level** *7 clear counters*
Step 5 **privilege exec all level** *level command-string*
Changes the privilege level of the reload command from privilege level 15 to privilege level 7.
Router(config)# **privilege exec all level** *7 reload*
Step 6 **end**
Exits global configuration mode.
Router(config)# **end**

The following example shows the enforcement of the settings above and privilege levels.

Step 1 **enable** *level password*
Logs the user into the networking device at the privilege level specified for the level argument.
Router> **enable** *7 Zy72sKj*
Step 2 **show privilege**
Displays the privilege level of the current CLI session
Router# **show privilege**
Current privilege level is 7
Step 3 **clear counters**
The clear counters command clears the interface counters. This command has been changed from privilege level 15 to privilege level 7.
Router# **clear counters**
Clear "show interface" counters on all interfaces [confirm]
Router#
02:41:37: %CLEAR-5-COUNTERS: Clear counter on all interfaces by console
Step 4 **clear ip route** *
The *ip route* argument string for the **clear** command should not be allowed because it was not changed from privilege level 15 to privilege level 7.
Router# **clear ip route ***
^
% Invalid input detected at '^' marker.
Router#
Step 5 **reload in time**
The reload command causes the networking device to reboot.
Router# **reload in** *10*
Reload scheduled in 10 minutes by console
Proceed with reload? [confirm]
Router#
***
*** --- SHUTDOWN in 0:10:00 ---
***
02:59:50: %SYS-5-SCHEDULED_RELOAD: Reload requested for 23:08:30 PST Sun Mar 20
Step 6 **reload cancel**
The reload cancel terminates a reload that was previously setup with the reload in time command.

Router# **reload cancel**
***
*** --- SHUTDOWN ABORTED ---
***
04:34:08: %SYS-5-SCHEDULED_RELOAD_CANCELLED: Scheduled reload cancelled at 15:38:46 PST
Sun Mar 27 2005

Step 7 **disable**

Exits the current privilege level and returns to privilege level 1.

Router# **disable**

Step 8 **show privilege**

Displays the privilege level of the current CLI session

Router> **show privilege**
Current privilege level is 1

The term “authorized administrator” is used in this ST to refer to any user that has been assigned to a privilege level that is permitted to perform the relevant action; therefore has the appropriate privileges to perform the requested functions. The privilege level determines the functions the user can perform; hence the authorized administrator with the appropriate privileges. Refer to the Guidance documentation and IOS Command Reference Guide for available commands and associated roles and privilege levels.

The Switch can and shall be configured to authenticate all access to the command line interface using a username and password.

7.5.2 Security function summary

The security function is designed to satisfy the following security functional requirements.

FMT_MTD.1

FMT_SMF.1

FMT_SMR.1

7.6 SF06: Protection of the TSF

The TOE implements several measures to protect the TSF.

7.6.1 Security function description

FPT_ITT.1(1), (2)

The TOE is self-contained and provides all of the claimed functionality within a single appliance. However if more than one TOE is used in the configuration, the TOE may be configured to use the cryptographic services as described in the FCS SFRs to secure the connection and protect the transmitted data. If the TOE is configured as a Virtual Switching System (VSS) the objectives for the operational environment must be enforced for the complete (distributed) system, i.e. both chassis and the link are assumed to be physically secured by the environment.

FPT_PTD_EXT.1(1) and FPT_PTD_EXT.1(2)

The TOE includes a Master Passphrase features that can be used to configure the TOE to encrypt all locally defined user passwords. In this manner, the TOE ensures that plaintext user passwords will not be disclosed even to administrators. Password encryption is configured using the ‘service password-encryption’ command.

The TOE stores all private keys in a secure directory that is not readily accessible to administrators. All pre-shared and symmetric keys are stored in encrypted form to using AES encryption to prevent access.

This functionality is configured on the TOE using the 'password encryption aes' command. The TOE is configured to not display configured keys as part of configuration files using the 'hidekeys' command.

FPT_RPL.1

By virtue of the cryptographic and path mechanisms implemented by the TOE, replayed network packets directed (terminated) at the TOE will be detected and discarded. Note: The intended scope of this requirement is trusted communications with the TOE (e.g., administrator to TOE, IT entity (e.g., authentication server) to TOE,). As such, replay does not apply to receipt of multiple network packets due to network congestion or lost packet acknowledgments.

FPT_STM.1

The TOE provides a source of date and time information for the switch, used in audit timestamps and in validating service requests. This function can only be accessed from within the configuration exec mode via the privileged mode of operation of the switch. The clock function is reliant on the system clock provided by the underlying hardware. The timestamp is assumed to be accurate to an official time source, such as Network Time Protocol (NTP) server. Therefore, the TOE can optionally be set to receive time from an NTP server. The NTP synchronizes the TOE clock to the U.S. Naval Observatory Master Clocks in Washington, DC and Colorado Springs CO. The NTP sends periodic requests and adjusts the clock as necessary. If an NTP server is used, the TOE supports signature verification of the timestamp from the time server.

FPT_TUD_EXT.1

The TOE has specific versions that can be queried by an administrator. When updates are made available by Cisco, an administrator can obtain and install those updates. The cryptographic checksums (i.e., public hashes) are used to verify software/firmware update files (to ensure they have not been modified from the originals distributed by Cisco) before they are used to actually update the applicable TOE components.

FPT_TST_EXT.1

As a FIPS 140-2 validated product, the TOE runs a suite of self-tests during initial start-up to verify its correct operation. Refer to the FIPS Security Policy for available options and management of the cryptographic self-test. For testing of the TSF, the TOE automatically runs checks and tests at startup and during resets to ensure the TOE is operating correctly.

7.6.2 Security function summary

The security function is designed to satisfy the following security functional requirements.

FPT_ITT.1(1)

FPT_ITT.1(2)

FPT_PTD_EXT.1(1)

FPT_PTD_EXT.1(2)

FPT_RPL.1

FPT_STM.1

FPT_TUD_EXT.1

FPT_TST_EXT.1

7.7 SF07: Resource utilization

The TOE implements several measures to protect usage of resources.

7.7.1 Security function description

FRU_RSA.1

An administrator can configure a maximum number of concurrent sessions for remote administrative interfaces.

7.7.2 Security function summary

The security function is designed to satisfy the following security functional requirements.

FRU_RSA.1

7.8 SF08: TOE access

The TOE implements access rules for external entities.

7.8.1 Security function description

FTA_SSL_EXT.1 and FTA_SSL.3

An administrator can configure maximum inactivity times for both local and remote administrative sessions. When a session is inactive (i.e., not session input) for the configured period of time the TOE will terminate the session, flush the screen, and no further activity is allowed requiring the administrator to log in (be successfully identified and authenticated) again to establish a new session. The allowable range is from 1 to 65535 seconds.

FTA_TAB.1

The TOE displays a privileged Administrator specified banner on the CLI management interface prior to allowing any administrative access to the TOE. This is applicable for both local and remote TOE administration.

7.8.2 Security function summary

The security function is designed to satisfy the following security functional requirements.

FTA_SSL_EXT.1

FTA_SSL.3

FTA_TAB.1

7.9 SF09: Trusted path/channels

The TOE provides secure communication channels to other trusted IT products.

7.9.1 Security function description

FTP_ITC.1(1) and FTP_ITC.1(2)

The TOE protects communications with peer or neighbour routers using key hash as defined in FCS_COP.1(4).1. This protects the data from disclosure by encryption and by checksums that verify that data has not been modified.

FTP_TRP.1(1) and FTP_TRP.1(2)

All remote administrative communications take place over a secure encrypted IPSec tunnel. The IPSec session is encrypted using AES encryption. The remote users are able to initiate IPSec communications with the TOE.

7.9.2 Security function summary

The security function is designed to satisfy the following security functional requirements.

FTP_ITC.1(1)

FTP_ITC.1(2)

FTP_TRP.1(1)

FTP_TRP.1(2)

8 Appendixes

The appendixes provide explanations for abbreviations and references to relevant standards.

8.1 Abbreviations

A.x	Assumption x on the environment
AAA	Administration, Authorization, and Accounting
ACL	Access Control List
ADV	CC assurance class development
ADV_ARC	CC assurance family security architecture
ADV_FSP	CC assurance family functional specification
ADV_TDS	CC assurance family TOE design
AES	Advanced Encryption Standard
AGD	CC assurance class guidance documents
AGD_OPE	CC assurance family operational user guidance
AGD_PRE	CC assurance family preparative procedures
ALC	CC assurance class life cycle support
ALC_CMC	CC assurance family CM capabilities
ALC_CMS	CC assurance family CM scope
ALC_DEL	CC assurance family delivery
ASE	CC assurance class security target evaluation
ASE_CCL	CC assurance family conformance claim
ASE_ECD	CC assurance family extended components definition
ASE_INT	CC assurance family ST introduction
ASE_OBJ	CC assurance family security objectives
ASE_REQ	CC assurance family security requirements
ASE_SPD	CC assurance family security problem definition
ASE_TSS	CC assurance family TOE summary specification
ATE	CC assurance class tests
ATE_COV	CC assurance family coverage
ATE_FUN	CC assurance family functional tests

	ATE_IND	CC assurance family independent testing
	AVA	CC assurance class vulnerability assessment
	AVA_VAN	CC assurance family vulnerability analysis
	BGP	Border Gateway Protocol. An exterior gateway protocol. It performs routing between multiple autonomous systems and exchanges routing and reachability information with other BGP systems.
	BSI	Bundesamt für Sicherheit in der Informationstechnik, Federal office for information security
2540	CC	Common Criteria
	CEM	Common Evaluation Methodology for Information Technology Security
	CLI	Command Line Interface
	CM	Configuration Management
	DH	Diffie-Hellman
	DSZ	Deutsches IT-Sicherheitszertifikat, German certificate for IT security
	EAL	Evaluation assurance level
	EAL2	EAL level 2, predefined package of CC
	EAL2+	EAL2 augmented by at least one SFR or SAR
	EDCS	Engineering Document Control System
2550	EIGRP	Enhanced Interior Gateway Routing Protocol (Cisco proprietary)
	FAU	CC functional class security audit
	FCS	CC functional class cryptographic support
	FDP	CC functional class user data protection
	FIA	CC functional class identification and authentication
	FIPS	Federal Information Processing Standard
	FMT	CC functional class security management
	FPT	CC functional class protection of the TSF
	FRU	CC functional class resource utilisation
	FTA	CC functional class TOE access
2560	FTP	CC functional class trusted path/channels
	HA	High Availability (device or component failover)
	HMAC	Hashed Message Authentication Code
	HSRP	Hot Standby Router Protocol (Cisco proprietary)
	HTTPS	Hyper-Text Transport Protocol Secure
	IEEE	Institute of Electrical and Electronics Engineers
	IGMP	Internet Group Management Protocol
	IOS	Cisco proprietary Internetwork Operating System
	IP	Internet Protocol
	IPSec	IP Security
2570	IT	Information technology
	MAC	Media Access Control
	NDPP	Network Device Protection Profile
	NEBS	Network Equipment-Building System
	NTP	Network Time Protocol
	OE.x	Security objective x for the environment
	OS	Operating System
	OSP	Organizational security policy
	OSPF	Open Shortest Path First. An interior gateway protocol (routes within a single autonomous system). A link-state routing protocol which calculates the shortest path
2580		to each node.
	O.x	Security objective x for the TOE
	P.x	OSP x enforced by environment
	PP	Protection profile

	PRNG	Pseudo Random Number Generator
	PVLAN	Private VLAN
	RADIUS	Remote Authentication Dial In User Service
	RIP	Routing Information Protocol
	RNG	Random Number Generator
	RSA	Rivest, Shamir and Adleman (algorithm for public-key cryptography)
2590	SAR	Security assurance requirement
	SFR	Security functional requirement
	SM	Service Module
	SPD	Security problem definition
	SSH	Secure Shell
	SSHv2	Secure Shell (version 2)
	ST	Security target
	Sup2T	Cisco Supervisor Engine 2T (VS-S2T-10G or VS-S2T-10G-XL)
	T.x	Threat x
	TACACS	Terminal Access Controller Access Control System
2600	TCP	Transport Control Protocol
	TCP/IP	Transmission Control Protocol/Internet Protocol
	TDES	Triple Data Encryption Standard
	TLS	Transport Layer Security
	TOE	Target of evaluation
	TSF	TOE security functionality
	TSFI	TSF interface
	TSP	TOE Security Policy
	UDP	User Datagram Protocol
	VACL	VLAN ACL
2610	VLAN	Virtual Local Area Network
	VSL	Virtual Switch Link
	VSS	Virtual Switching System

8.2 References

	CC	Common Criteria for Information Technology Security Evaluation, Part 1 to Part 3, July 2009, Version 3.1, Revision 3, Final
	CC-P1	Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and general model, July 2009, Version 3.1, Revision 3, Final, CCMB-2009-07-001
	CC-P2	Common Criteria for Information Technology Security Evaluation, Part 2: Security functional components, July 2009, Version 3.1, Revision 3, Final, CCMB-2009-07-002
2620	CC-P3	Common Criteria for Information Technology Security Evaluation, Part 3: Security assurance components, July 2009, Version 3.1, Revision 3, Final, CCMB-2009-07-003
	NDPP	Network Device Protection Profile (NDPP), Version 1.0, December 10, 2010
	KS2011	W. Killmann, W. Schindler, "A proposal for: Functionality classes for random number generators", Version 2.0, September 18, 2011