Security Requirements for Network Devices

Information Assurance Directorate

10 December 2010

Version 1.0

Table of Contents

List of Tables

Revision History

Version	Date	Description
1.0	10 December 2010	Initial release

1 INTRODUCTION

This Protection Profile (PP), describing security requirements for a Network Device (defined to be an infrastructure device that can be connected to a network), is intended to provide a minimal, baseline set of requirements that are targeted at mitigating well defined and described threats. It represents an evolution of "traditional" Protection Profiles and the associated evaluation of the requirements contained within the document. This introduction will describe the features of a compliant TOE, and will also discuss the evolutionary aspects of the PP as a guide to readers of the document.

1.1 Compliant Targets of Evaluation

This is a Protection Profile (PP) for a network device. A network device in the context of this PP is a device composed of hardware and software that is connected to the network and has an infrastructure role in the overall enterprise. Examples of a "network device" that should claim compliance to this PP include routers, firewalls, IDSs, audit servers, and switches that have Layer 3 functionality. Examples of devices that connect to a network but are not suitable for evaluation against this PP include mobile devices ("smart phones"), end-user workstations, SQL servers, web servers, application servers, and database servers. Devices operating at Layer 1 and/or Layer 2 such as hubs, bridges, and access switches are suitable for evaluation against this PP if they have a network interface used for remote administration.

While the functionality that the TOE is obligated to implement (in response to the described threat environment) is discussed in detail in later sections, it is useful to give a brief description here. Compliant TOEs will provide security functionality that addresses threats to the TOE and implements policies that are imposed by law or regulation. Compliant TOEs must protect communications to and between elements of a distributed TOE (e.g., between a network IDS sensor and the centralized IDS manager) or instantiations of the TOE in a single enterprise (e.g., between routers). The TOE must offer identification and authentication services that support the composition of moderate complex passwords or passphrases, and make these services available locally (that is, a local logon) as well as remotely (remote login). The TOE must also offer auditing of a set of events that are associated with security-relevant activity on the TOE, although these events will be stored on a device that is distinct from the TOE. The TOE must offer some protection for common network denial of service attacks and must also provide the ability to verify the source of updates to the TOE.

While the protocols required by this PP make use of certificates, this version of the PP does not levy requirements on the certificate infrastructure (for example, using OCSP to verify a certificate's validity). Such requirements will be included in future versions of this document.

It is intended that the set of requirements in this PP is limited in scope in order to promote quicker, less costly evaluations that provide some value to end users. STs that include a large amount of additional functionality (and requirements) are discouraged. Future modules will be used to specify

sets of additional functionality (e.g., Firewalls, VPNs), which can then be used by ST writers looking to specify additional functionality.

2 SECURITY PROBLEM DESCRIPTION

As detailed in the previous section, the security problem to be addressed by compliant TOEs is described by threats and policies that are common to network devices, as opposed to those that might be targeted at the specific functionality of a specific type of network device. Annex A: Supporting Tables presents the Security Problem Description (SPD) in a more "traditional" form. The following sections detail the problems that compliant TOEs will address; references to the "traditional" statements in Annex A are included.

2.1 Communications with the TOE

Network devices communicate with other network devices, as well as administrators, over the network. The endpoints of the communication can be both geographically and logically distant from the TOE, and pass through a variety of other systems. These intermediate systems may be under the control of the adversary, and offer an opportunity for communications with the TOE to be compromised. While these communications fall into three distinct categories (the TOE communicating with a remote administrator; the TOE communicating in a distributed processing environment with another instance or instances of itself; and the TOE communicating with another IT entity that is not another instance of the TOE (e.g., an NTP server or a peer router)), the threats to the communication between these endpoints are the same.

Plaintext communication with the TOE may allow critical data (such as passwords, configuration settings, and routing updates) to be read and/or manipulated directly by intermediate systems, leading to a compromise of the TOE. Several protocols can be used to provide protection; however, each of these protocols have myriad options that can be implemented and still have the overall protocol implementation remain compliant to the protocol specification listed in the RFC. Some of these options can have negative impacts on the security of the connection. For instance, using a weak encryption algorithm (even one that is allowed by the RFC, such as DES) can allow an adversary to read and even manipulate the data on the encrypted channel, thus circumventing countermeasures in place to prevent such attacks. Further, if the protocol is implemented with little-used or non-standard options, it may be compliant with the protocol specification but will not be able to interact with other, diverse equipment that is typically found in large enterprises.

Even though the communication path is protected, there is a possibility that the external user (be it a remote administrator, another instance of the distributed TOE, or a trusted IT entity such as a peer router) could be duped into thinking that a malicious third-party user or system is the TOE. For instance, a middleman could intercept a connection request to the TOE, and respond to the external user as if it were the TOE. In a similar manner, the TOE could also be duped into thinking that it is establishing communications with a legitimate remote entity when in fact it is not. An

attacker could also mount a malicious man-in-the-middle-type of attack, in which an intermediate system is compromised, and the traffic is proxied, examined, and modified by this system. This attack can even be mounted via encrypted communication channels if appropriate countermeasures are not applied. These attacks are, in part, enabled by a malicious attacker capturing network traffic (for instance, an authentication session) and "playing back" that traffic in order to fool an endpoint into thinking it was communicating with a legitimate remote entity.

[T.UNAUTHORIZED_ACCESS]

2.2 Malicious "Updates"

Since the most common attack vector used involves attacking unpatched versions of software containing well-known flaws, updating network device firmware is necessary to ensure that changes to threat environment are addressed. Timely application of patches ensures that the system is a "hard target", thus increasing the likelihood that product will be able to maintain and enforce its security policy. However, the updates to be applied to the product must be trustable in some manner; otherwise, an attacker can write their own "update" that instead contains malicious code of their choosing, such as a rootkit, bot, or other malware. Once this "update" is installed, the attacker then has control of the system and all of its data.

Methods of countering this threat typically involve hashes of the updates, and potentially cryptographic operations (e.g., digital signatures) on those hashes as well. However, the validity of these methods introduces additional threats. For instance, a weak hash function could result in the attacker being able to modify the legitimate update in such a way that the hash remained unchanged. For cryptographic signature schemes, there are dependencies on

1) the strength of the cryptographic algorithm used to provide the signature, and

2) the ability of the end user to verify the signature (which typically involves checking a hierarchy of digital signatures back to a root of trust (a certificate authority)).

If a cryptographic signature scheme is weak, then it may be compromised by an attacker and the end user will install a malicious update, thinking that it is legitimate. Similarly, if the root of trust can be compromised, then a strong digital signature algorithm will not stop the malicious update from being installed (the attacker will just create their own signature on the update using the compromised root of trust, and the malicious update will then be installed without detection).

[T.UNAUTHORIZED_UPDATE]

2.3 Undetected System Activity

While several threats are directed at specific capabilities of the TOE, there is also the threat that activity that could indicate an impending or on-going security compromise could go undetected.

Administrators can unintentionally perform actions on the TOE that compromise the security being provided by the TOE; for instance, a mis-configuration of security parameters. Processing performed in response to user data (for example, the establishment of a secure communications session, cryptographic processing associated with a protected session) may give indications of a failure or compromise of a TOE security mechanism (e.g., establishment of a session with an IT entity when no such sessions should be taking place). When indications of activity that may impact the security of the TOE are not generated and monitored, it is possible for harmful activity to take place on the TOE without responsible officials being aware and able to correct the problem. Further, if no data are kept or records generated, reconstruction of the TOE and the ability to understand the extent of any compromise could be negatively affected.

While this PP requires that the TOE generates the audit data, these data are not required to be stored on the TOE, but rather sent to a trusted external IT entity (e.g., a *syslog* server). These data may be read or altered by an intervening system, thus potentially masking indicators of suspicious activity. It may also be the case that the TOE could lose connectivity to the external IT entity, meaning that the audit information could not be sent to the repository.

[T.ADMIN_ERROR, T.UNDETECTED_ACTIONS, T.UNAUTHORIZED_ACCESS]

2.4 Accessing the TOE

In addition to the threats discussed in Section 2.1 dealing with the TOE communicating with various external parties that focus on the communications themselves, there are also threats that arise from attempts to access the TOE, or the means by which these access attempts are accomplished.

For example, if the TOE does not discriminate between users that are allowed to access the TOE interactively (through a locally connected console, or with a session-oriented protocol such as SSH) and a user with no authority to use the TOE in this manner, the configuration of the TOE cannot be trusted. Assuming that there is this distinction, there is still the threat that one of the allowed accounts may be compromised and used by an attacker that does not otherwise have access to the TOE.

One vector for such an attack is the use of poor passwords by authorized administrators of the TOE. Passwords that are too short, are easily-guessed dictionary words, or are not changed very often, are susceptible to a brute force attack. Additionally, if the password is plainly visible for a period of time (such as when a legitimate user is typing it in during logon) then it might be obtained by an observer and used to illegitimately access the system.

Once a legitimate user is logged on, there still are a number of threats that need to be considered. During the password change process, if the TOE does not verify that it is the user associated with the account changing the password, then anyone can change the password on a legitimate account and take that account over. If a user walks away from a logged-in session, then another person with no access to the device could sit down and illegitimately start accessing the TOE.

[T.UNAUTHORIZED_ACCESS]

2.5 Resource Exhaustion

An ever increasing attack against devices with a network interface is a denial of service (DoS) attack. There are some resources that are common to all of the devices and require consideration; these are the resources that support the administrative interface. For example, if an attacker can fill up the connection table, an administrator could not remotely connect to the device's network interface to perform management functions.

Other resources may be able to be attacked; however, this is somewhat dependent on the nature of the device. For example, a switch that only operates up to level 2 on the network stack would not have the same DoS concerns as an application-filtering firewall that operates at a higher level in the network stack. Other considerations may include how much memory a device has, the amount of processing power, buffer allocations, etc., available to the device and the possibility for an attacker to render the device unable to perform its functions by illegitimately consuming resources.

[T.RESOURCE_EXHAUSTION]

2.6 User Data Disclosure

While most of the threats contained in this PP deal with TSF and administrative data, there is also a threat against user data that all network devices should mitigate. Data traversing the TOE could inadvertently be sent to a different user; since these data may be sensitive, this may cause a compromise that is unacceptable. The specific threat that must be addressed concerns user data that is retained by the TOE in the course of processing network traffic that could be inadvertently re-used in sending network traffic to a user other than that intended by the sender of the original network traffic.

[T.USER_DATA_REUSE]

2.7 TSF Failure

Security mechanisms of the TOE generally build up from a primitive set of mechanisms (e.g., memory management, privileged modes of process execution) to more complex sets of mechanisms. Failure of the primitive mechanisms could lead to a compromise in more complex mechanisms, resulting in a compromise of the TSF.

[T.TSF_FAILURE]

3 SECURITY OBJECTIVES

Compliant TOEs will provide security functionality that address threats to the TOE and implement policies that are imposed by law or regulation. The following sections provide a description of this functionality in light of the threats previously discussed that motivate its inclusion in compliant TOEs. The security functionality provided includes protected communications to and between elements of the TOE; administrative access to the TOE and its configuration capabilities; system monitoring for detection of security relevant events; control of resource availability; and the ability to verify the source of updates to the TOE.

3.1 Protected Communications

To address the issues concerning transmitting sensitive data to and from the TOE described in Section 2.1, "Communications with the TOE", compliant TOEs will provide encryption for these communication paths between themselves and the endpoint. These channels are implemented using one (or more) of three standard protocols: IPsec, TLS/HTTPS, and SSH. These protocols are specified by RFCs that offer a variety of implementation choices. Requirements have been imposed on some of these choices (particularly those for cryptographic primitives) to provide interoperability and resistance to cryptographic attack. While compliant TOEs must support all of the specified choices, they may support additional algorithms and protocols. Whether such additional mechanisms will be evaluated is Scheme-dependent. If such additional mechanisms are not evaluated, guidance must be given to the administrator so that they can be disabled (or shown not to affect the specified security functionality) during TOE operation.

In addition to providing protection from disclosure (and detection of modification) for the communications, each of the protocols described in this document (IPsec, SSH, and TLS/HTTPS) offer two-way authentication of each endpoint in a cryptographically secure manner, meaning that even if there was a malicious attacker between the two endpoints, any attempt to represent themselves to either endpoint of the communications path as the other communicating party would be detected. The requirements on each protocol, in addition to the structure of the protocols themselves, provide protection against replay attacks such as those described in Section 2.1, usually by including a unique value in each communication so that replay of that communication can be detected.

(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, FCS_SSH_EXT.1, FCS_TLS_EXT.1, FCS_HTTPS_EXT.1), FPT_PTD.1(2), FPT_ITT.1(1), FPT_ITT.1(2), FPT_RPL.1, FTP_ITC.1(1), FTP_ITC.1(2), FTP_TRP.1(1), FTP_TRP.1(2))

3.2 Verifiable Updates

As outlined in Section 2.2, "Malicious Updates", failure by the Security Administrator to verify that updates to the system can be trusted may lead to compromise of the entire system. A first step in

establishing trust in the update is to publish a hash of the update that can be verified by the System Administrator prior to installing the update. In this way, the Security Administrator can download the update, compute the hash, and compare it to the published hash. While this establishes that the update downloaded is the one associated with the published hash, it does not indicate if the source of the update/hash combination has been compromised or can't be trusted. So, there remains a threat to the system. To establish trust in the source of the updates, the system can provide cryptographic mechanisms and procedures to procure the update, check the update cryptographically through the TOE-provided digital signature mechanism, and install the update on the system. While there is no requirement that this process be completely automated, administrative guidance documentation will detail any procedures that must be performed manually, as well as the manner in which the administrator ensures that the signature on the update is valid.

(FPT_TUD_EXT.1, FCS_COP.1(2), FCS_COP.1(3))

3.3 System Monitoring

In order to assure that information exists that allows Security Administrators to discover intentional and unintentional issues with the configuration and/or operation of the system as discussed in Section 2.3, "Undetected System Activity", compliant TOEs have the capability of generating audit data targeted at detecting such activity. Auditing of administrative activities provides information that may hasten corrective action should the system be configured incorrectly. Audit of select system events can provide an indication of failure of critical portions of the TOE (e.g., a cryptographic provider process not running) or anomalous activity (e.g., establishment of an administrative session at a suspicious time, repeated failures to establish sessions or authenticate to the system) of a suspicious nature.

In some instances there may be a large amount of audit information produced that could overwhelm the TOE or administrators in charge of reviewing the audit information. The TOE must be capable of sending audit information to an external trusted entity, which mitigates the possibility that the generated audit data will cause some kind of denial of service situation on the TOE. This information must carry reliable timestamps, which will help order the information when sent to the external device.

Loss of communication with the audit server is problematic. While there are several potential mitigations to this threat, this PP does not mandate that a specific action takes place; the degree to which this action preserves the audit information and still allows the TOE to meet its functionality responsibilities should drive decisions on the suitability of the TOE in a particular environment.

(FAU_GEN.1, FAU_GEN.2, FAU_STG_EXT.1, FAU_STG_EXT.3, FPT_STM.1)

3.4 TOE Administration

In order to provide a trusted means for administrators to interact with the TOE, the TOE provides a password-based logon mechanism. The administrator must have the capability to compose a strong password, and have mechanisms in place so that the password must be changed regularly. To avoid attacks where an attacker might observe a password being typed by an administrator, passwords must be obscured during logon. Session locking or termination must also be implemented to mitigate the risk of an account being used illegitimately.

(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.1(1), FTA_SSL_EXT.1, FTA_SSL.3)

3.5 Resource Availability

Denial of service attacks, as discussed in Section 2.6, “Resource Exhaustion”, can impact the ability of the TOE to perform its operational and security functions. In order to mitigate the effect of such attacks, quotas are placed on the amount of exhaustible resources that can be allocated. These quotas enable the TOE to reserve a portion of the resources so that the device can remain operational, and the effects of such attacks can be addressed by administrative personnel.

(FRU_RSA)

3.6 Residual Information Clearing

In order to counter the threat that user data is inadvertently included in network traffic not intended by the original sender, the TSF ensures that network packets sent from the TOE do not include data "left over" from the processing of previous network information.

(FDP_RIP.2)

3.7 TSF Self Test

In order to detect some number of failures of underlying security mechanisms used by the TSF, the TSF will perform self-tests. The extent of this self testing is left to the product developer, but a more comprehensive set of self tests should result in a more trustworthy platform on which to develop enterprise architecture.

(FPT_TST_EXT.1)

4 SECURITY REQUIREMENTS

The Security Functional Requirements included in this section are derived from Part 2 of the *Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 3*, with additional extended functional components.

4.1 Conventions

The CC defines operations on Security Functional Requirements: assignments, selections, assignments within selections and refinements. This document uses the following font conventions to identify the operations defined by the CC:

- Assignment: Indicated with *italicized* text;
- Refinement made by PP author: Indicated with **bold** text and strikethroughs, if necessary;
- Selection: Indicated with underlined text;
- Assignment within a Selection: Indicated with *italicized and underlined* text;
- Iteration: Indicated by appending the iteration number in parenthesis, e.g., (1), (2), (3).

Explicitly stated SFRs are identified by having a label 'EXT' after the requirement name for TOE SFRs.

4.2 TOE Security Functional Requirements

This section identifies the Security Functional Requirements for the TOE. The TOE Security Functional Requirements that appear below in Table 1 are described in more detail in the following subsections.

Table 1: TOE Security Functional Requirements and 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.
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.	

Requirement	Auditable Events	Additional Audit Record Contents
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.1(1)	None.	
FPT_PTD.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.	
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.

4.2.1 Security Audit (FAU)

FAU_GEN.1 Audit Data Generation

FAU_GEN.1.1 The TSF shall be able to generate an audit record of the following auditable events:

a) Start-up and 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 1*].

Application Note: The ST author can include other auditable events directly in the table; they are not limited to the list presented.

Many auditable aspects of the SFRs included in this document deal with administrative actions. Item c above requires all administrative actions to be auditable, so no additional specification of the auditability of these actions is specified in Table 1.

Assurance Activity:
The evaluator shall check the administrative guide and ensure that it lists all of the auditable events and provides a format for audit records. Each audit record format type must be covered, along with a brief description of each field. The evaluator shall check to make sure that every audit event type mandated by the 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.

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.

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.

FAU_GEN.1.2 The TSF shall record within each audit record at least the following information:

a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and
b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [*information specified in column three of Table 1*].

Application Note: As with the previous component, the ST author should update Table 1 above with any additional information generated. "Subject identity" in the context of this requirement could either be the administrator's user id or the affected network interface, for example.

FAU_GEN.2 User Identity Association

FAU_GEN.2.1 For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

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: 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.

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

FAU_STG.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: 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.

Assurance Activity:
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.

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.

4.2.2 Cryptographic Support (FCS)

FCS_CKM.1 Cryptographic Key Generation (for asymmetric keys)

FCS_CKM.1.1 **Refinement:** The TSF shall generate **asymmetric** cryptographic keys **in accordance with a domain parameter generator and** [**selection: (1) a random number generator, and/or (2) a prime 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.**

Application Note: The generated key strength of 2048-bit DSA and rDSA keys need to be equivalent to, or greater than, a symmetric key strength of 112 bits. See NIST Special Publication 800-57, "Recommendation for Key Management" for information about equivalent key strengths.

b) Case: For domain parameters used in finite field-based key establishment schemes[1]

- **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"**

[1] For example, "classic" Diffie-Hellman-based scheme.

- **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: This component requires that the TOE be able to generate the public/private key pairs that are used for the digital signature operations in FCS_COP.1(2). If multiple schemes are supported, then the ST author should iterate this requirement and FCS_COP.1(2) to capture this capability.

Assurance Activity:
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.

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: "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.

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

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

FCS_COP.1.1(1) **Refinement:** The TSF shall perform [*encryption and decryption*] in accordance with a specified cryptographic algorithm [*AES operating in* [**assignment: *one or more modes***]] and cryptographic key sizes 128-bits, 256-bits, and [**selection: 192 bits, no other key sizes**] that meets the following:

- **FIPS PUB 197, "Advanced Encryption Standard (AES)"**
- **[Selection: NIST SP 800-38A, NIST SP 800-38B, NIST SP 800-38C, NIST SP 800-38D, NIST SP 800-38E]**

Application Note: For the assignment, the ST author should choose the mode or modes in which AES operates. For the first selection, the ST author should choose the key sizes that are supported by this functionality. For the second selection, the ST author should choose the standards that describe the modes specified in the assignment.

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

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

FCS_COP.1.1(2) **Refinement:** The TSF shall perform **cryptographic signature services** in accordance with a ***[selection:***

(1) Digital Signature Algorithm (DSA) with a key size (modulus) of 2048 bits or greater,

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

(3) Elliptic Curve Digital Signature Algorithm (ECDSA) with a key size of 256 bits or greater]

Application Note: As the preferred approach for cryptographic signature, elliptic curves will be required in future publications of this PP.

that meets the following:

Case: Digital Signature Algorithm

- **[selection: FIPS PUB 186-3, "Digital Signature Standard" , FIPS PUB 186-2, "Digital Signature Standard"]**

Case: RSA Digital Signature Algorithm

- **[selection: FIPS PUB 186-3, "Digital Signature Standard" , FIPS PUB 186-2, "Digital Signature Standard"]**

Case: Elliptic Curve Digital Signature Algorithm

- **[selection: FIPS PUB 186-3, "Digital Signature Standard" , FIPS PUB 186-2, "Digital Signature Standard"]**
- **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 ST Author should choose the algorithm implemented to perform digital signatures; if more than one algorithm is available, this requirement (and the corresponding FCS_CKM.1 requirement) should be iterated to specify the functionality. For the algorithm chosen, the ST author should make the appropriate assignments/selections to specify the parameters that are implemented for that algorithm.

While FIPS PUB 186-2 has been revised by FIPS PUB 186-3, it is still allowable to claim conformance to the older standard while products are transitioning to the newer standard. At a future date, products will not be allowed to claim conformance to FIPS PUB 186-2. The ST author makes the selection of the conformance standard as appropriate for the TOE.

For elliptic curve-based schemes, the key size refers to the log_2 of the order of the base point. As the preferred approach for digital signatures, ECDSA will be required in future publications of this PP.

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

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

FCS_COP.1.1(3) **Refinement:** The TSF shall perform [*cryptographic hashing services*] in accordance with a specified cryptographic algorithm [**selection: SHA-1, SHA-256, SHA-384, SHA-512] and message digest sizes** [**selection: 160, 256, 384, 512**] **bits** that meet the following: *FIPS Pub 180-3, "Secure Hash Standard."*

Application Note: The selection of the hashing algorithm must correspond to the selection of the message digest size; for example, if SHA-1 is chosen, then the only valid message digest size selection would be 160 bits.

In subsequent publications of this PP, it is likely that SHA-1 will no longer be an approved algorithm for cryptographic hashing.

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

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

FCS_COP.1.1(4) **Refinement:** The TSF shall perform [*keyed-hash message authentication*] in accordance with a specified cryptographic algorithm HMAC-[**selection: SHA-1, SHA-256, SHA-384, SHA-512], key size [assignment: *key size (in bits) used in HMAC*], and message digest sizes** [**selection: 160, 256, 384, 512**] **bits** that meet the following: *FIPS Pub 198-1, "The Keyed-Hash Message Authentication Code, and FIPS Pub 180-3, "Secure Hash Standard."*

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

Extended: Cryptographic Operation (Random Bit Generation) (FCS_RBG_(EXT))

FCS_RBG_(EXT).1 Extended: Cryptographic Operation (Random Bit Generation)

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: 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.

Assurance Activity:
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.

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.

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.

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: 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.

4.2.3 User Data Protection (FDP)

FDP_RIP.2 Full Residual Information Protection

FDP_RIP.2.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the [selection: *allocation of the resource to, deallocation of the resource from*] all objects.

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

4.2.4 Identification and Authentication (FIA)

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: 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 previous password.*

Application Note: 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.

Assurance Activity:
The evaluator shall examine the operational guidance to determine that it provides guidance to security administrators on the composition of strong passwords, and that it provides instructions on setting the minimum password length; the 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. 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.

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: 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).

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: 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.

FIA_UAU.6 Re-authenticating

FIA_UAU.6.1 The TSF shall re-authenticate the user under the conditions: *when the user changes their password, [selection: following TSF-initiated locking (FTA_SSL), [assignment: other conditions], no other conditions].*

Assurance Activity:
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.

FIA_UAU.7 Protected Authentication Feedback

FIA_UAU.7.1 The TSF shall provide only *obscured feedback* to the user while the authentication is in progress at the local console.

Application Note: "Obscured feedback" implies the TSF does not produce a visible display of any authentication data entered by a user (such as the echoing of a password), although an obscured indication of progress may be provided (such as an asterisk for each character). It also implies that the TSF does not return any information during the authentication process to the user that may provide any indication of the authentication data.

4.2.5 Security Management (FMT)

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

FMT_MTD.1.1 The TSF shall restrict the ability to *manage* the *TSF data* to the *Security Administrators*.

Application Note: The word "manage" includes but is not limited to create, initialize, view, change default, modify, delete, clear, and append. This requirement is intended to be the "default" requirement for management of TSF data; other iterations of FMT_MTD should place different restrictions or operations available on the specifically-identified TSF data. TSF data includes cryptographic information as well; managing these data would include the association of a cryptographic protocol with an interface, for instance.

FMT_SMF.1 Specification of Management Functions

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.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 [selection: no other functions, [assignment: other cryptographic functions (or other functions) used to support the update capability]]*

Application Note: At a minimum the TOE must provide the functionality for the administrator to verify that the update received came from a trusted source; this is done using digital signatures. If other mechanisms are used, those should be specified in the assignment; otherwise "no other functions" should be selected. If the other mechanisms used are cryptographic, then the ST author should ensure they are specified using FCS components, and that those components are referenced in the assignment.

FMT_SMR.1 Security Roles

FMT_SMR.1.1 The TSF shall maintain the roles:

- *[Security Administrator*,
- *Selection:[assignment: other administrative roles], No other roles]]*.

Application Note: PP authors should add required roles as appropriate for the given technology or capability (e.g., inclusion of an audit capability may require the creation of an "auditor" role). FMT_MOF, FMT_MTD, and FMT_MSA requirements should be added to reflect the capabilities of these roles in such cases.

FMT_SMR.1.2 The TSF shall be able to associate users with roles.

4.2.6 Protection of the TSF (FPT)

FPT_ITT.1(1) Basic Internal TSF Data Transfer Protection (Disclosure)

FPT_ITT.1.1(1) **Refinement:** 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: [assignment: *FCS-specified service used to protect TSF data from disclosure]***.

Application Note: The ST author includes a reference to the applicable cryptographic service into the assignment statement (e.g., if IPsec is used, then referring to FCS_IPSEC_EXT would be sufficient).

FPT_ITT.1(2) Basic Internal TSF Data Transfer Protection (Modification)

FPT_ITT.1.1(2) **Refinement:** The TSF shall **detect modification of** TSF data when it is transmitted between separate parts of the TOE **through the use of the TSF-provided cryptographic services: [assignment: *FCS-specified service used to detect modification of TSF data]***.

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: 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.

Assurance Activity:
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.

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: 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.

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

FPT_RPL.1 Replay Detection

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.

Application Note: The intent of the first element is that communications of a trusted nature (administrator to TOE, IT entity to TOE, TOE to TOE) are covered by the element and not subject to replay attacks.

FPT_STM.1 Reliable Time Stamps

FPT_STM.1.1 The TSF shall be able to provide reliable time stamps for its own use.

Extended: Trusted Update (FPT_TUD_(EXT).1)

FPT_TUD_(EXT).1 Extended: Trusted Update

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: 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.

Assurance Activity:
Updates to the TOE either have a hash associated with them, or are signed by an authorized source. If digital signatures are used, the definition of an authorized source is contained in the TSS, along with a description of how the certificates used by the update verification mechanism are contained on the device. The evaluator ensures this information is contained in the TSS. The evaluator also

ensures that the TSS (or the operational guidance) describes how the candidate updates are obtained; the processing associated with verifying the digital signature or calculating the hash of the updates; and the actions that take place for successful (hash or signature was verified) and unsuccessful (hash or signature could not be verified) cases. The evaluator shall perform the following tests:

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

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

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.

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

4.2.7 Resource Utilization (FRU)

FRU_RSA.1 Maximum Quotas

FRU_RSA.1.1 The TSF shall enforce maximum quotas of the following resources: [assignment: *resources supporting the administrative interface*], [selection: [assignment: *controlled resources*], *no other resource*] that [selection: *individual user, defined group of users, subjects*] can use [selection: *simultaneously, over a specified period of time*].

Application Note: At a minimum, compliant TOEs must impose quotas on exhaustible resources used to support the remote administrative interface; these are listed in the first assignment. Other resources that can be controlled (e.g., TCP connection resources) should be listed in the second assignment; if there are no other resources then the last item in the selection should be chosen. The second selection should be chosen to reflect the consumers of the resource that are to be controlled.

The last selection is used to limit the timeframe associated with the use of the controlled resources (e.g., a quota on the number of TCP connection requests from a given IP address in 30 seconds).

4.2.8 TOE Access (FTA)

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.

Assurance Activity:
The evaluator shall perform the following test:

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

FTA_SSL.3 TSF-initiated Termination

FTA_SSL.3.1 **Refinement:** The TSF shall terminate **a remote** interactive session after a [*Security Administrator-configurable time interval of session inactivity*].

Assurance Activity:
The evaluator shall perform the following test:

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

FTA_TAB.1 Default TOE Access Banners

FTA_TAB.1.1 **Refinement:** Before establishing **a user/administrator** session the TSF shall display **a Security Administrator-specified** advisory **notice and consent** warning message regarding unauthorized use of the TOE.

Application Note: This requirement is intended to apply to interactive sessions between a human user and a TOE. IT entities establishing connections or programmatic connections (e.g., remote procedure calls over a network) are not required to be covered by this requirement.

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

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

4.2.9 Trusted Path/Channels (FTP)

FTP_ITC.1(1) Inter-TSF Trusted Channel (Prevention of Disclosure)

FTP_ITC.1.1(1) **Refinement:** The TSF shall **use [assignment: *FCS-specified service*] to** provide a **trusted** communication channel between itself and **authorized IT entities** that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from disclosure.

FTP_ITC.1.2(1) **Refinement:** The TSF shall permit *the TSF*, ***or the authorized IT entities*** to initiate communication via the trusted channel.

FTP_ITC.1.3(1) The TSF shall initiate communication via the trusted channel for [*all authentication functions*, [assignment: *protected communications/protocols between peers*]].

Application Note: The ST author should fill in the assignment with the name of the protocol used to establish the trusted channel. This requirement addresses the case where the TOE establishes communications with an IT peer (VPN, router updates, etc.).

FTP_ITC.1(2) Inter-TSF Trusted Channel (Detection of Modification)

FTP_ITC.1.1(2) **Refinement:** The TSF shall **use [assignment: *FCS-specified service*] in providing** a **trusted** communication channel between itself and **authorized IT entities** that is logically distinct

from other communication channels and provides assured identification of its end points and **detection of the modification of data**.

FTP_ITC.1.2(2) **Refinement:** The TSF shall permit *the TSF*, ***or the authorized IT entities*** to initiate communication via the trusted channel.

FTP_ITC.1.3(2) The TSF shall initiate communication via the trusted channel for [all authentication functions, [assignment: *protected communications/protocols between peers*]].

Application Note: The ST author should fill in the assignment with the name of the protocol used to establish the trusted channel. This requirement addresses the case where the TOE establishes communications with an IT peer (VPN, router updates, etc.).

FTP_TRP.1(1) Trusted Path

FTP_TRP.1.1(1) **Refinement:** The TSF shall provide a communication path between itself and *remote administrators* **using [assignment: *FCS-specified service*]** 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.1.2(1) The TSF shall permit *remote administrators* to initiate communication via the trusted path.

FTP_TRP.1.3(1) **Refinement:** The TSF shall require the use of the trusted path for *all remote administrative actions*.

FTP_TRP.1(2) Trusted Path

FTP_TRP.1.1(2) **Refinement:** The TSF shall provide a communication path between itself and *remote administrators* **using [assignment: *FCS-specified service*]** that is logically distinct from other communication paths and provides assured identification of its end points and **detection of modification of the communicated data**.

Application Note: The refinement is necessary because it is not required (and in most cases impractical) for the TSF to prevent the data from being modified; it is sufficient to detect this occurrence.

FTP_TRP.1.2(2) The TSF shall permit *remote administrators* to initiate communication via the trusted path.

FTP_TRP.1.3(2) **Refinement:** The TSF shall require the use of the trusted path for *all remote administrative actions*.

4.3 Security Assurance Requirements

The Security Objectives for the TOE in Section 3 were constructed to address threats identified in Section 2. The Security Functional Requirements (SFRs) in Section 4.2 are a formal instantiation of the Security Objectives. The PP draws from EAL1 the Security Assurance Requirements (SARs) to frame the extent to which the evaluator assesses the documentation applicable for the evaluation and performs independent testing.

While this section contains the complete set of SARs from the CC, the Assurance Activities to be performed by an evaluator are detailed both in Section 4.2 as well as in this section.

The general model for evaluating TOEs against STs written to conform to this PP is as follows:

After the ST has been approved for evaluation, the Common Criteria Testing Laboratory (CCTL) will obtain the TOE, supporting IT environment, and the administrative guides for the TOE. The Assurance Activities listed in the ST (which will be refined by the CCTL to be TOE-specific, either within the ST or in a separate document) will then be performed by the CCTL. The CCTL is also expected to perform all of the actions mandated by the Common Evaluation Methodology (CEM) for EAL1. The results of these activities will be documented and presented (along with the administrative guidance used) for validation.

For each assurance family, "Developer Notes" are provided on the developer action elements to clarify what, if any, additional documentation/activity needs to be provided by the developer. For the content/presentation and evaluator activity elements, additional assurance activities (to those already contained in Section 4.2 and the CEM for EAL1) are described as a whole for the family, rather than for each element. Additionally, the assurance activities described in this section are complementary to those specified in Section 4.2.

The TOE security assurance requirements, summarized in Table 2, identify the management and evaluative activities required to address the threats identified in Section 2 of this PP.

Table 2: TOE Security Assurance Requirements

Assurance Class	Assurance Components	Assurance Components Description
Development	ADV_FSP.1	Basic Functional Specification
Guidance Documents	AGD_OPE.1	Operational user guidance
	AGD_PRE.1	Preparative User guidance
Tests	ATE_IND.1	Independent testing - conformance
Vulnerability Assessment	AVA_VAN.1	Vulnerability analysis
Life Cycle Support	ALC_CMC.1	Labeling of the TOE
	ALC_CMS.1	TOE CM coverage

4.3.1 Class ADV: Development

At EAL1, the information about the TOE is contained in the guidance documentation available to the end user as well as the TOE Summary Specification (TSS) portion of the ST. While it is not required that the TOE developer write the TSS, the TOE developer must concur with the description of the product that is contained in the TSS as it relates to the functional requirements. The Assurance Activities contained in Section 4.2, coupled with the CEM, should provide the ST authors with sufficient information to determine the appropriate content for the TSS section.

4.3.1.1 ADV_FSP.1 Basic Functional Specification

The functional specification describes the Target Security Functions Interfaces (TSFIs). At EAL1, it is not necessary to have a formal or complete specification of these interfaces. Additionally, because TOEs conforming to this PP will necessarily have interfaces to the Operational Environment that are not directly invokable by TOE users, at EAL1 there is little point specifying that such interfaces be described in and of themselves since only indirect testing of such interfaces may be possible. For this PP, the activities for this family should focus on understanding the interfaces presented in the TSS in response to the functional requirements and the interfaces presented in the AGD documentation. No additional "functional specification" documentation is necessary to satisfy the assurance activities specified.

The interfaces that need to be evaluated are characterized through the information needed to perform the assurance activities listed, rather than as an independent, abstract list.

Developer action elements:

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.
Developer Note:	As indicated in the introduction to this section, the functional specification is comprised of the information contained in the AGD_OPE and AGD_PRE documentation, coupled with the information provided in the TSS of the ST. The assurance activities in the functional requirements point to evidence that should exist in the documentation and TSS section; since these are directly associated with the SFRs, the tracing in element ADV_FSP.1.2D is implicitly already done and no additional documentation is necessary.

Content and presentation elements:

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 categorization of interfaces as SFR-non-interfering.

ADV_FSP.1.4C	The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification. **Evaluator action elements:**
ADV_ FSP.1.1E	The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.
ADV_ FSP.1.2E	The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

Assurance Activity:

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

4.3.2 Class AGD: Guidance Documents

The guidance documents will be provided with the developer's security target. Guidance must include a description of how the authorized user verifies that the Operational Environment can fulfill its role for the security functionality. The documentation should be in an informal style and readable by an authorized user.

Guidance must be provided for every operational environment that the product supports as claimed in the ST. This guidance includes

- instructions to successfully install the TOE in that environment; and
- instructions to manage the security of the TOE as a product and as a component of the larger operational environment.

Guidance pertaining to particular security functionality is also provided; specific requirements on such guidance are contained in the assurance activities specified in Section 4.2.

4.3.2.1 AGD_OPE.1 Operational User Guidance

	Developer action elements:
AGD_OPE.1.1D	The developer shall provide operational user guidance.
Developer Note:	Rather than repeat information here, the developer should review the assurance activities for this component to ascertain the specifics of the guidance that the evaluator will be checking for. This will provide the necessary information for the preparation of acceptable guidance. **Content and presentation elements:**
AGD_OPE.1.1C	The operational user guidance shall describe, for each user role, the user-

accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.

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

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

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

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

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

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

Evaluator action elements:

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

Assurance Activity:

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

The operational guidance shall at a minimum list the processes running (or that could run) on the TOE in its evaluated configuration during its operation that are capable of processing data received on the network interfaces (there are likely more than one of these, and this is not limited to the process that "listens" on the network interface). It is acceptable to list all processes running (or that could run) on the TOE in its evaluated configuration instead of attempting to determine just those that process the network data. For each process listed, the administrative guidance will contain a short (e.g., one- or two-line) description of the process' function, and the privilege with which the service runs. "Privilege" includes the hardware privilege level (e.g., ring 0, ring 1), any software privileges specifically associated with the process, and the privileges associated with the user role the process runs as or under.

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

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

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

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

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

4.3.2.2 AGD_PRE.1 Preparative Procedures

Developer action elements:

AGD_PRE.1.1D The developer shall provide the TOE, including its preparative procedures.

Developer Note: As with the operational guidance, the developer should look to the assurance activities to determine the required content with respect to preparative procedures.

Content and presentation elements:

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.

Evaluator action elements:

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

AGD_ PRE.1.2E The evaluator *shall apply* the preparative procedures to confirm that the TOE can be prepared securely for operation.

Assurance Activity:

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

4.3.3 Class ATE: Tests

Testing is specified for functional aspects of the system as well as aspects that take advantage of design or implementation weaknesses. The former is done through the ATE_IND family, while the latter is through the AVA_VAN family. At the assurance level specified in this PP, testing is based on advertised functionality and interfaces with dependency on the availability of design information. One of the primary outputs of the evaluation process is the test report as specified in the following requirements.

4.3.3.1 ATE_IND.1 Independent Testing - Conformance

Testing is performed to confirm the functionality described in the TSS as well as the administrative (including configuration and operational) documentation provided. The focus of the testing is to confirm that the requirements specified in Section 4.2 are being met, although some additional testing is specified for SARs in Section 4.3. The Assurance Activities identify the additional testing activities associated with these components. The evaluator produces a test report documenting the plan for and results of testing, as well as coverage arguments focused on the platform/TOE combinations that are claiming conformance to this PP.

Developer action elements:

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

Content and presentation elements:

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

Evaluator action elements:

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

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

Assurance Activity:

The evaluator shall prepare a test plan and report documenting the testing aspects of the system. The test plan covers all of the testing actions contained in the CEM and the body of 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.

4.3.4 Class AVA: Vulnerability Assessment

For the first generation of this protection profile, the evaluation lab is expected to survey open sources to discover what vulnerabilities have been discovered in these types of products. In most cases, these vulnerabilities will require sophistication beyond that of a basic attacker. Until penetration tools are created and uniformly distributed to the evaluation labs, the evaluator will not be expected to test for these vulnerabilities in the TOE. The labs will be expected to comment on the likelihood of these vulnerabilities given the documentation provided by the vendor. This information will be used in the development of penetration testing tools and for the development of future protection profiles.

4.3.4.1 AVA_VAN.1 Vulnerability Survey

Developer action elements:

AVA_VAN.1.1D The developer shall provide the TOE for testing.

Content and presentation elements:

AVA_VAN.1.1C The TOE shall be suitable for testing.

Evaluator action elements:

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

AVA_VAN.1.2E The evaluator *shall perform* a search of public domain sources to identify potential vulnerabilities in the TOE.

AVA_VAN.1.3E The evaluator *shall conduct* penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Basic attack potential.

Assurance Activity:
As with ATE_IND, the evaluator shall generate a report to document their findings with respect to this requirement. This report could physically be part of the overall test report mentioned in ATE_IND, or a separate document. The evaluator performs a search of public information to determine the vulnerabilities that have been found in network infrastructure devices 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.

4.3.5 Class ALC: Life-cycle Support

At the assurance level provided for TOEs conformant to this PP, life-cycle support is limited to end-user-visible aspects of the life-cycle, rather than an examination of the TOE vendor's development and configuration management process. This is not meant to diminish the critical role that a developer's practices play in contributing to the overall trustworthiness of a product; rather, it's a reflection on the information to be made available for evaluation at this assurance level.

4.3.5.1 ALC_CMC.1 Labeling of the TOE

This component is targeted at identifying the TOE such that it can be distinguished from other products or versions from the same vendor and can be easily specified when being procured by an end user.

Developer action elements:

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

Content and presentation elements:

ALC_CMC.1.1C The TOE shall be labeled with its unique reference.

Evaluator action elements:

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

Assurance Activity:
The evaluator shall check the ST to ensure that it contains an identifier (such as a product name/version number) that specifically identifies the version that meets the requirements of the ST.

Further, the evaluator shall check the AGD guidance and TOE samples received for testing to ensure that the version number is consistent with that in the ST. If the vendor maintains a web site advertising the TOE, the evaluator shall examine the information on the web site to ensure that the information in the ST is sufficient to distinguish the product.

4.3.5.2 ALC_CMS.1 TOE CM Coverage

Given the scope of the TOE and its associated evaluation evidence requirements, this component's assurance activities are covered by the assurance activities listed for ALC_CMC.1.

	Developer action elements:
ALC_CMS.2.1D	The developer shall provide a configuration list for the TOE.
	Content and presentation elements:
ALC_CMS.2.1C	The configuration list shall include the following: the TOE itself; and the evaluation evidence required by the SARs.
ALC_CMS.2.2C	The configuration list shall uniquely identify the configuration items.
	Evaluator action elements:
ALC_CMS.2.1E	The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

Assurance Activity:
The "evaluation evidence required by the SARs" in 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.

RATIONALE

The rationale tracing the threats to the objectives and the objectives to the requirements is contained in the prose in Sections 2.0 and 3.0. The only outstanding mappings are those for the Assumptions and Organizational Security Policies; those are contained in Annex A below.

ANNEX A: SUPPORTING TABLES

In this Protection Profile, the focus in the initial sections of the document is to use a narrative presentation in an attempt to increase the overall understandability of the threats to network devices; the methods used to mitigate those threats; and the extent of the mitigation achieved by compliant TOEs. This presentation style does not readily lend itself to a formalized evaluation activity, so this Annex contains the tabular artifacts that can be used for the evaluation activities associated with this document.

Assumptions

The specific conditions listed in the following subsections are assumed to exist in the TOE's Operational Environment. These assumptions include both practical realities in the development of the TOE security requirements and the essential environmental conditions on the use of the TOE.

PP authors should ensure that the assumptions still hold for their particular technology; the table should be modified as appropriate.

Table 3: TOE Assumptions

Assumption Name	Assumption Definition
A.NO_GENERAL_PURPOSE	It is assumed that there are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE.
A.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is assumed to be provided by the environment.
A.TRUSTED_ADMIN	TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.

Threats

The following threats should be integrated into the threats that are specific to the technology by the PP authors when including the requirements described in this document. Modifications, omissions, and additions to the requirements may impact this list, so the PP author should modify or delete these threats as appropriate.

Table 4: Threats

Threat Name	Threat Definition
T.ADMIN_ERROR	An administrator may unintentionally install or configure the TOE incorrectly, resulting in ineffective security mechanisms.
T.RESOURCE_EXHAUSTION	A process or user may deny access to TOE services by exhausting critical resources on the TOE.
T.TSF_FAILURE	Security mechanisms of the TOE may fail, leading to a compromise of the TSF.
T.UNDETECTED_ACTIONS	Malicious remote users or external IT entities may take actions that adversely affect the security of the TOE. These actions may remain undetected and thus their effects cannot be effectively mitigated.
T.UNAUTHORIZED_ACCESS	A user may gain unauthorized access to the TOE data and TOE executable code. A malicious user, process, or external IT entity may masquerade as an authorized entity in order to gain unauthorized access to data or TOE resources. A malicious user, process, or external IT entity may misrepresent itself as the TOE to obtain identification and authentication data.
T.UNAUTHORIZED_UPDATE	A malicious party attempts to supply the end user with an update to the product that may compromise the security features of the TOE.
T.USER_DATA_REUSE	User data may be inadvertently sent to a destination not intended by the original sender.

Organizational Security Policies

An organizational security policy is a set of rules, practices, and procedures imposed by an organization to address its security needs. PP Authors should ensure that any policies that apply to their particular technology are captured in the following table, and that the policies listed below are applicable.

Table 5: Organizational Security Policies

Policy Name	Policy Definition
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.

Security Objectives for the TOE

Table 6: Security Objectives for the TOE

TOE Security Obj.	TOE Security Objective Definition
O.PROTECTED_COMMUNICATIONS	The TOE will provide protected communication channels for administrators, other parts of a distributed TOE, and

TOE Security Obj.	TOE Security Objective Definition
	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.

The following table contains objectives for the Operational Environment. As assumptions are added to the PP, these objectives should be augmented to reflect such additions.

Table 7: Security Objectives for the Operational Environment

TOE Security Obj.	TOE Security Objective Definition
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.

ANNEX B: NIST SP 800-53/CNSS 1253 MAPPING

Several of the NIST SP 800-53/CNSS 1253 controls are either fully or partially addressed by compliant TOEs. This section outlines the requirements that are addressed, and can be used by certification personnel to determine what, if any, additional testing is required when the TOE is incorporated into its operational configuration.

Application Note: In this version, only a simple mapping is provided. In future versions, additional narrative will be included that will provide further information for the certification team. This additional information will include details regarding the SFR to control mapping discussing what degree of compliance is provided by the TOE (e.g., fully satisfies the control, partially satisfies the control). In addition, a comprehensive review of the specified assurance activities, and those evaluation activities that occur as part of satisfying the SARs will be summarized to provide the certification team information regarding how compliance was determined (e.g., document review, vendor assertion, degree of testing/verification). This information will indicate to the certification team what, if any, additional activities they need to perform to determine the degree of compliance to specified controls.

Since the ST will make choices as far as selections, and will be filling in assignments, a final story cannot necessarily be made until the ST is complete and evaluated. Therefore, this information should be included in the ST in addition to the PP. Additionally, there may be some necessary interpretation (e.g.,"modification") to the activities performed by the evaluator based on a specific implementation. The scheme could have the oversight personnel (e.g., Validators) fill in this type of information, or could have this done by the evaluator as part of the assurance activities. The verification activities are a critical piece of information that must be provided so the certification team can determine what, if anything, they need to do in addition to the work of the evaluation team.

Identifier	Name	Applicable SFRs
AC-3	Access Enforcement	FMT_MTD.1
AC-6	Least Privilege	FMT_MTD.1
AC-8	System use Notification	FTA_TAB.1
AC-11	Session Lock	FIA_UAU.6, FTA_SSL_EXT.1
AC-14	Permitted Actions Without Identification	FIA_UIA_EXT.1
AC-17(7)	Remote Access	FCS_SSH_EXT.1
AU-2	Auditable Events	FAU_GEN.1
AU-2(4)		FAU_GEN.1
AU-3	Content of Audit Records	FAU_GEN.1, FAU_GEN.2
AU-3(1)		FAU_GEN.1
AU-5	Response to Audit Processing Failures	FAU_STG_EXT.3
AU-8	Time Stamps	FPT_STM.1
AU-10	Non-Repudiation	FCS_COP.1(2)
AU-12	Audit Generation	FAU_GEN.1
IA-2	Identification and Authentication	FIA_UIA_EXT.1, FIA_UAU_EXT.5
IA-5	Authenticator Management	FIA_PMG_EXT.1
IA-6	Authenticator Feedback	FIA_UAU.7

SC-4	Information in Shared Resources	FDP_RIP.2
SC-6	Resource Priority	FRU_RSA.1
SC-8	Transmission Integrity	FCS_IPSEC_EXT.1, FCS_TLS_EXT.1, FCS_HTTPS_EXT.1, FCS_SSH_EXT.1, FPT_ITT.1(2), FTP_ITC.1(2)
SC-9	Transmission Confidentiality	FCS_IPSEC_EXT.1, FCS_TLS_EXT.1, FCS_HTTPS_EXT.1, FCS_SSH_EXT.1, FPT_ITT.1(1), FTP_ITC.1(1)
SC-10	Network Disconnect	FTA_SSL.3
SC-11	Trusted Path	FTP_TRP.1 (1), FTP_TRP.1(2)
SC-12	Cryptographic Key Establishment and Management	FCS_CKM.1, FCS_CKM_EXT.4
SI-6	Security Functionality Verification	FPT_TST_EXT.1

ANNEX C: ADDITIONAL REQUIREMENTS

As indicated in the body of this PP, there are several methods by which conformant TOEs can mitigate threats against compromise of the communication channel between administrators, other portions of the (distributed) TOE, or external IT entities. Either IPsec or SSH must be implemented by the TOE; however, it is also allowable for both to be implemented, and TLS/HTTPS may also be implemented in addition to one or both of IPsec and SSH. Since there are requirements associated with each of the protocol suites, specification of the protocols in the PP becomes confusing and problematic, since specification of optional requirements is not readily supported by the CC. In order to address this situation as cleanly as possible, the following requirements should be included in the ST depending on the selections for the FCS_COMM_PROT_EXT.1 component.

Note that minor adjustments to the narrative information in the beginning of the ST may be required depending on the selections performed. Additionally, depending on the requirements selected, the appropriate information from Section C1.2 *Auditable Events* will need to be added to the auditable events table in the ST.

C1.1 Requirements

FCS_IPSEC_EXT.1 Explicit: IPSEC

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: 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.

Assurance Activity:
The evaluator shall examine the TSS to verify that it describes how "confidentiality only" ESP mode is disabled. 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.

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. 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.

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: 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.

Assurance Activity:
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. 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.

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: 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.

Assurance Activity:
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. 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.

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 Note: 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.

Assurance Activity:
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. 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.

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: The selected algorithm should correspond to an appropriate selection for FCS_COP.1(2).

Assurance Activity:
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. 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.

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.

Assurance Activity:
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 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. 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.

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: 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.

Assurance Activity:
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. 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.

FCS_TLS_EXT.1 Explicit: TLS

FCS_TLS_EXT.1.1 The TSF shall implement one or more of the following protocols [selection: TLS 1.0 (RFC 2346), TLS 1.1 (RFC 4346), TLS 1.2 (RFC 5246)] supporting the following ciphersuites:

Mandatory Ciphersuites:
TLS_RSA_WITH_AES_128_CBC_SHA
TLS_RSA_WITH_AES_256_CBC_SHA
TLS_DHE_RSA_WITH_AES_128_CBC_SHA
TLS_DHE_RSA_WITH_AES_256_CBC_SHA

Optional Ciphersuites:
[selection:
None
TLS_RSA_WITH_AES_128_CBC_SHA256
TLS_RSA_WITH_AES_256_CBC_ SHA256
TLS_DHE_RSA_WITH_AES_128_CBC_ SHA256
TLS_DHE_RSA_WITH_AES_256_CBC_ SHA256
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
].

Application Note: The ST author must make the appropriate selections and assignments to reflect the TLS 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.

The ciphersuites to be used in the evaluated configuration are limited by this requirement. The ST author should select the optional ciphersuites that are supported; if there are no ciphersuites supported other than the mandatory suites, then "None" should be selected. If administrative steps need to be taken so that the suites negotiated by the implementation are limited to those in this requirement, the appropriate instructions need to be contained in the guidance called for by AGD_OPE.

The Suite B algorithms (RFC 5430) listed above are the preferred algorithms for implementation. Future publications of this PP will require support for TLS 1.2 (RFC 5246). In addition, future publications of this PP will require that the TOE offer a means to deny all connection attempts using specified older versions of the SSL/TLS protocol.

Assurance Activity:
The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that optional characteristics (e.g., extensions supported, client authentication supported) are specified, and the ciphersuites supported are specified as well. The evaluator shall check the TSS to ensure that the ciphersuites specified are identical to those listed for this component. The evaluator

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

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

FCS_SSH_EXT.1 Explicit: SSH

FCS_SSH_EXT.1.1 The TSF shall implement the SSH protocol that complies with RFCs 4251, 4252, 4253, and 4254.

Application Note: 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.

FCS_SSH_EXT.1.2 The TSF shall ensure that the SSH connection be rekeyed after no more than 2^{28} packets have been transmitted using that key.

Assurance Activity:
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.

FCS_SSH_EXT.1.3 The TSF shall ensure that the SSH protocol implements a timeout period for authentication as defined in RFC 4252 of [*assignment: timeout period*], and provide a limit to the number of failed authentication attempts a client may perform in a single session to [*assignment: maximum number of attempts*] attempts.

Application Note: In the first assignment, the ST author should insert the timeout period (e.g., "10 minutes") from the initiation of authentication session after which the session should timeout if authentication has been unsuccessful. In the second assignment, the maximum number of failed authentication attempts is specified. The RFC indicates the server should drop the session after this number of failed attempts.

Assurance Activity:
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. 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.

FCS_SSH_EXT.1.4 The TSF shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based, password-based.

Assurance Activity:
The evaluator shall check to ensure that the TSS contains a description of the public key algorithms that are acceptable for use for authentication, that this list conforms to FCS_SSH_EXT.1.7, and ensure that password-based authentication methods are also allowed. The evaluator shall also perform the following tests:

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

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

FCS_SSH_EXT.1.5 The TSF shall ensure that, as described in RFC 4253, packets greater than [*assignment: number of bytes*] bytes in an SSH transport connection are dropped.

Application Note: RFC 4253 provides for the acceptance of "large packets" with the caveat that the packets should be of "reasonable length" or dropped. The assignment should be filled in by the ST author with the maximum packet size accepted, thus defining "reasonable length" for the TOE.

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

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

FCS_SSH_EXT.1.6 The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms: AES-CBC-128, AES-CBC-256, [selection: AEAD_AES_128_GCM, AEAD_AES_256_GCM, *no other algorithms*].

Application Note: In subsequent publications of this PP, it is likely that AES-GCM will be required and CBC will become optional. In the assignment, the ST author can select the AES-GCM algorithms, or "no other algorithms" if AES-GCM is not supported. If AES-GCM is selected, there should be corresponding FCS_COP entries in the ST.

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

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

FCS_SSH_EXT.1.7 The TSF shall ensure that the SSH transport implementation uses SSH_RSA and [selection: *PGP-SIGN-RSA, PGP-SIGN-DSS, no other public key algorithms,*] as its public key algorithm(s).

Application Note: RFC 4253 specifies required and allowable public key algorithms. This requirement makes SSH-RSA "required" and allows two others to be claimed in the ST. The ST author should make the appropriate selection, selecting "no other public key algorithms" if only SSH_RSA is implemented.

Assurance Activity:
The assurance activity associated with FCS_SSH_EXT.1.4 verifies this requirement.

FCS_SSH_EXT.1.8 The TSF shall ensure that data integrity algorithms used in SSH transport connection is [*selection: hmac-sha1, hmac-sha1-96, hmac-md5, hmac-md5-96*].

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

FCS_SSH_EXT.1.9 The TSF shall ensure that diffie-hellman-group14-sha1 is the only allowed key exchange method used for the SSH protocol.

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

- Test 1: The evaluator shall attempt to perform a diffie-hellman-group1-sha1 key exchange, and observe that the attempt fails. The evaluator shall then attempt to perform a diffie-hellman-group14-sha1 key exchange, and observe that the attempt succeeds.

FCS_HTTPS_EXT.1 Explicit: HTTPS

FCS_HTTPS_EXT.1.1 The TSF shall implement the HTTPS protocol that complies with RFC 2818.

Application Note: 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.

FCS_HTTPS_EXT.1.2 The TSF shall implement HTTPS using TLS as specified in FCS_TLS_EXT.1.

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

C1.2 Auditable Events

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.

Requirement	Auditable Events	Additional Audit Record Contents
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.
FCS_TLS_EXT.1	Failure to establish a TLS Session. Establishment/Termination of a TLS session.	Reason for failure. Non-TOE endpoint of connection (IP address) for both successes and failures.
FCS_SSH_EXT.1	Failure to establish an SSH session Establishment/Termination of an SSH session	Reason for failure Non-TOE endpoint of connection (IP address) for both successes and failures.
FCS_HTTPS_EXT.1	Failure to establish a HTTPS Session. Establishment/Termination of a HTTPS session.	Reason for failure. Non-TOE endpoint of connection (IP address) for both successes and failures.