PP-Module for SSL/TLS Inspection Proxies

Version: 1.1
2021-09-10
National Information Assurance Partnership

Revision History

Version	Date	Comment
1.1	2021-09-10	Updates to reflect Github conversion, compatibility with NDcPP v2.2E, and Technical Decisions applied to version 1.0
1.0	2019-08-23	Update release

Contents

1 Introduction

1.1 Overview

The scope of this PP-Module is to describe the security functionality of an SSL/TLS Inspection Proxy (STIP) in terms of [CC] and to define functional and assurance requirements for such products. This PP-Module is intended for use with the following Base-PPs:

- collaborative Protection Profile for Netowrk Devices (NDcPP), Version 2.2E

This Base-PP is valid because a STIP is a specific type of network appliance that is able to function as an authorized man-in-the-middle for TLS connections.

This PP-Module is intended to specify the functionality of a network device that includes limited Certification Authority (CA) functionality to issue certificates for the purpose of providing network security services on the underlying plaintext. The device accomplishes this by terminating an intended TLS session between a monitored client and specified external servers. The device instead establishes a TLS session thread consisting of a TLS session between the device and the external server and a second TLS session between the device, acting as the external server, and the client. By replacing the end-to-end TLS session with two TLS sessions terminated at the TOE, the device is able to provide additional security services based on the decrypted plaintext.

A network device meeting this PP-Module may perform additional security services on the plaintext, provide the decrypted payload to external network devices to perform the security services, or do both. These additional security services, whether processed internally or externally, may be performed inline, or passively. If multiple security services are provided, some may be inline, while others are performed passively. This PP-Module does not cover the specific requirements associated with various additional services.

A TOE that conforms to a PP-Configuration containing this PP-Module may be a ‘Distributed TOE’ as defined in the NDcPP; however, the SSL/TLS Inspection Proxy functionality described in this PP-Module should be in a single TOE component. This PP-Module does not prohibit the TOE from implementing other security functionality in a distributed manner.

1.2 Terms

The following sections list Common Criteria and technology terms used in this document.

1.2.1 Common Criteria Terms

Assurance	Grounds for confidence that a TOE meets the SFRs [CC].
Base Protection Profile (Base-PP)	Protection Profile used as a basis to build a PP-Configuration.
Common Criteria (CC)	Common Criteria for Information Technology Security Evaluation (International Standard ISO/IEC 15408).
Common Criteria Testing Laboratory	Within the context of the Common Criteria Evaluation and Validation Scheme (CCEVS), an IT security evaluation facility, accredited by the National Voluntary Laboratory Accreditation Program (NVLAP) and approved by the NIAP Validation Body to conduct Common Criteria-based evaluations.
Common Evaluation Methodology (CEM)	Common Evaluation Methodology for Information Technology Security Evaluation.
Distributed TOE	A TOE composed of multiple components operating as a logical whole.
Operational Environment (OE)	Hardware and software that are outside the TOE boundary that support the TOE functionality and security policy.
Protection Profile (PP)	An implementation-independent set of security requirements for a category of products.
Protection Profile Configuration (PP-Configuration)	A comprehensive set of security requirements for a product type that consists of at least one Base-PP and at least one PP-Module.

Protection Profile Module (PP-Module)	An implementation-independent statement of security needs for a TOE type complementary to one or more Base Protection Profiles.
Security Assurance Requirement (SAR)	A requirement to assure the security of the TOE.
Security Functional Requirement (SFR)	A requirement for security enforcement by the TOE.
Security Target (ST)	A set of implementation-dependent security requirements for a specific product.
TOE Security Functionality (TSF)	The security functionality of the product under evaluation.
TOE Summary Specification (TSS)	A description of how a TOE satisfies the SFRs in an ST.
Target of Evaluation (TOE)	The product under evaluation.

1.2.2 Technical Terms

Attribute	A characterization of an entity (monitored client or the server requested by a monitored client) used in the TLS session establishment policy or the plaintext processing policy implemented by the TOE that describes the entity. Common attributes include IP address, name, and certificates associated to an entity.
Block operation	A high-level operation of the TLS session establishment policy implemented by the TOE that prevents TLS sessions between a monitored client and the server requested by the client.
Bypass operation	A high-level operation of the TLS session establishment policy implemented by the TOE that allows a TLS session between a monitored client and the server requested by the client. Alternatively, an operation of the plaintext processing policy implemented by the TOE to bypass certain inspection processing functional components for plaintext data flows established under the SSL/TLS session establishment policy.
Inspect operation	A high-level operation of the TLS session establishment policy implemented by the TOE that establishes a TLS session thread between a monitored client and a server requested by the monitored client in order to provide security services on the underlying plaintext application data.
Inspection processing functional components	A discrete set of security functions implemented within a single logical component, internal or external to the TOE that provides security services based on a plaintext data flow controlled by the TOE intended to protect a monitored client from defined security threats, or to enforce a defined policy regarding the servers allowed to be accessed by monitored clients.
Monitored Client	A TLS client that uses the TOE as an SSL/TLS Inspection Proxy. This device requires a trust anchor to be installed for the internal CA of the TOE, and makes SSL/TLS requests for services external to the enclave. This client makes SSL/TLS requests to a “requested server” through the TOE.
Requested Server	The target of an SSL/TLS request by a monitored client through the TOE. It is typically a service provider for clients using SSL/TLS. If mutual authentication is to be supported, this device requires a trust anchor to be installed for the internal CA of the TOE.
Secure Sockets Layer/Transport Layer Security (SSL/TLS)	A set of security protocols defined by IETF RFCs to establish a secure point-to-point channel between a client and a server. The secure channel provides confidentiality, integrity and proof of origin to plaintext application data transferred between the client and server. SSL refers to early implementations of the SSL/TLS protocols that are deprecated. TLS refers to current versions of the SSL/TLS protocol.
TLS messages	Specific messages defined by TLS protocol standards. The TLS messages addressed in this PP-Module include TLS handshake messages: Client Hello, Server Hello, Server

	Certificate, Server Key Exchange, Client Key Exchange, Certificate Request, Client Certificate, Client Certificate Verify, Server Finished and Client Finished messages.
TLS session parameters	The parameters of a TLS session established by the TOE for protecting thrutraffic, minimally to include: the negotiated version, negotiated cipher suite, the size of any key exchange values sent or received in key exchange messages, the server certificate received, (a reference to) the server certificate sent, the client certificate received, (a reference to) the client certificate sent, and other negotiated values determined by the TLS handshake that are not fixed for all TLS sessions established.
TLS session thread	A connection negotiated by the TOE consisting of a TLS secure point-to-point channel between a monitored client and the TOE, a TLS secure point-to-point channel between the TOE and the requested server, and any traffic flow containing the underlying application plaintext decrypted from one of the SSL/TLS channels, that is transferred within or between inspection processing functional components controlled by the TOE.

1.3 Compliant Targets of Evaluation

The Target of Evaluation (TOE) may be a single device or a collection of devices that interact with each other to meet the requirements of this PP-Module. Other network devices can be used to supplement inspection of plaintext traffic made available by the TOE. Such external devices will be considered as part of the operational environment, unless they are used to meet the requirements of this PP-Module. Audit, web, or directory servers providing access to certificate validity information generated by the TOE, and intermediate or root certification authorities that issue certificates to the TOE’s embedded certification authority are considered part of the operational environment and external to the TOE, but interfaces to these essential services which are required for operation of the TOE will be considered within the TOE boundary. Assurance activities to validate an interface include inspection and exercise of these interfaces using a specific instance of the service (audit server, web server, and external certification authority) implemented within the test environment.

This PP-Module includes some functionality typical of firewalls. In particular, a device meeting this PP-Module is configurable so that it can block or process TLS traffic between monitored clients and requested servers. It’s important to note that the device may support TLS connections for remote administration; these TLS connections are distinct from those between the monitored clients and requested servers, and must meet different requirements. In the case of an SSL/TLS inspection proxy, the primary processing is to inspect the TLS traffic. A TOE also has the capability of passing the TLS handshake messages intact to allow end-to-end TLS encrypted traffic between monitored clients and specific servers without providing additional services (bypass the inspection) on decrypted traffic. The decision to drop, process, or bypass traffic is based on IP addresses and ports, as well as on the content of TLS handshake messages, including the certificate of the server, and other characteristics of the traffic that might be available. A device can also determine which additional security services, especially those provided by external network devices, are applied to a particular session based on the plaintext exposed, such as HTTP headers including uniform resource locators (URLs), user passwords, or other sensitive information.

This PP-Module does not require facilitating inspection of mutually authenticated TLS sessions. It does not address the management of clients required to support inspection, nor requirements to avoid monitored clients from discovering the existence of such inspection. Processing to support Certificate Pinning is included as an optional requirement since establishing an inspection point prevents the monitored clients from doing so themselves. Similarly, management of the TOE’s certificate trust store is required, since monitored clients cannot block traffic from sites using certificates issued by compromised CA certificates after the traffic is inspected.

1.3.1 TOE Boundary

A STIP is one or more network devices that uses CA functionality to replace an end-to-end TLS session with a TLS session between the STIP and a monitored client and another TLS session between the STIP and the TLS endpoint requested by the monitored client (the requested server). Additional functionality within the same network component as STIP functionality, or via external network devices, can be used to perform network security services, such as performing intrusion detection or providing reputation services on the plaintext traffic made available by the TOE. This functionality, while enabled by the TOE is out of scope. However, protecting and separating traffic flows of plaintext to or between discrete functional components performing such network security services is required and considered within the TOE. If the TOE provides an external interface to plaintext traffic for additional network security services, the entirety of all external processing will be considered a single functional component – the TOE is not responsible for controlling the flow of traffic among external systems.

All functionality described by the SFRs are within the TOE boundary, as is the ability for the TSF to establish secure remote connections with trusted entities in the Operational Environment (OE).

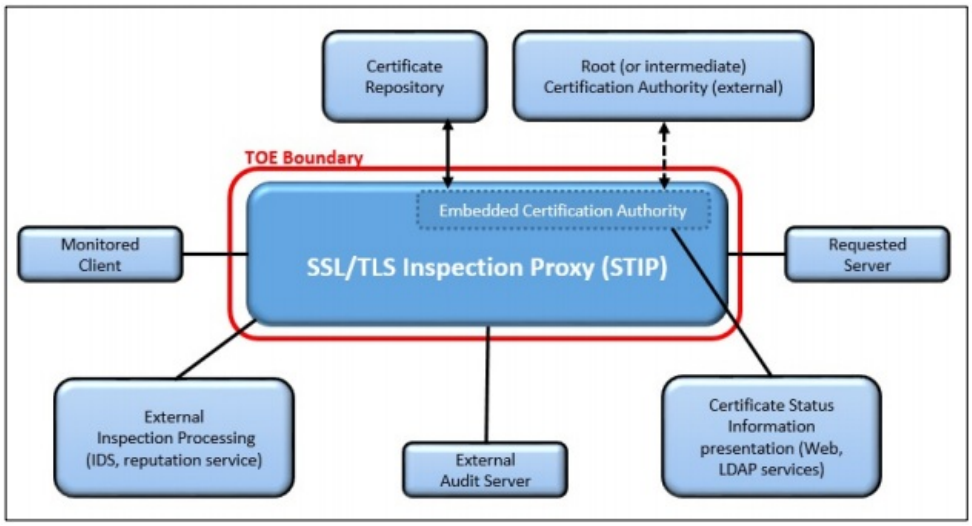


Figure 1: TLS Inspection Infrastructure

As can be seen from this figure, the TOE sits between a monitored client and requested server in order to intercept TLS traffic between them. For connections subject to inspection, the TOE will replace the endto-end TLS session between the monitored client and requested server and establish a TLS session thread in order to forward the plaintext application traffic to one or more inspection processing functional components in the operational environment for inspection. The TSF provides an embedded CA that is used to reconstruct the TLS channel and pass it to its intended destination in an encrypted format. The embedded CA provides certificates it issues to an (external) certificate repository and provides certificate status information to an (internal or external) certificate status presentation mechanism.

1.4 Use Cases

Requirements in this PP-Module are designed to address the security problem in the following use cases. The description of these use cases provide instructions for how the TOE and its OE should be made to support the functionality required by this PP-Module.

This PP-Module permits the inspection of mutually-authenticated TLS sessions between monitored clients and requested servers via exception processing. However, as a best practice, it is recommended instead that this behavior be handled as part of the TLS Inspection Bypass and/or TLS Session Blocking functionality. If the TOE provides inspection processing for mutually authenticated traffic, the ST must claim these optional SFRs.

This PP-Module does not specify routing policies for non-TLS traffic and exception processing should not be used to address functionality otherwise included in the collaborative Protection Profile for Stateful Traffic Filter Firewalls.

[USE CASE 1] Inspection Operation

The TOE intercepts traffic authorized for inspection from monitored clients requesting a serveronly authenticated TLS session with a requested server. The TOE initiates a TLS session with the requested server and validates the requested server’s certificate as legitimately issued by a trusted element of its trust store. The TOE authenticates the server on behalf of the client and generates a certificate that indicates the TOE is an authorized proxy for the requested server. The certificate is issued by the TOE’s embedded CA, which is trusted by the monitored client. The TOE establishes a valid TLS session with the monitored client using the issued certificate. Any TLS traffic between the monitored client and the requested server is decrypted by the TOE and assigned to a unique TLS session thread that is routed to one or more inspection processes, and encrypted into the respective TLS sessions.

[USE CASE 2] TLS Bypass Operation

The TOE intercepts traffic authorized for inspection from monitored clients requesting a serveronly authenticated TLS session with a requested server. The TOE initiates a TLS session with the requested server and validates the requested server’s certificate as legitimately issued by a trusted element of its trust store. The TOE authenticates the server on behalf of the client and generates a certificate that indicates the TOE is an authorized proxy for the requested server. The certificate is issued by the TOE’s embedded CA, which is trusted by the monitored client. The TOE establishes a valid TLS session with the monitored client using the issued certificate. Any TLS traffic between the monitored client and the requested server is decrypted by the TOE and assigned to a unique TLS session thread that is routed to one or more inspection processes, and encrypted into the respective TLS sessions.

[USE CASE 3] TLS Blocking Operation

The TOE intercepts traffic authorized for inspection from monitored clients requesting a serveronly authenticated TLS session with a requested server. The TOE initiates a TLS session with the requested server and validates the requested server’s certificate as legitimately issued by a trusted element of its trust store. The TOE authenticates the server on behalf of the client and generates a certificate that indicates the TOE is an authorized proxy for the requested server. The certificate is issued by the TOE’s embedded CA, which is trusted by the monitored client. The TOE establishes a valid TLS session with the monitored client using the issued certificate. Any TLS traffic between the monitored client and the requested server is decrypted by the TOE and assigned to a unique TLS session thread that is routed to one or more inspection processes, and encrypted into the respective TLS sessions.

[USE CASE 4] Exception Processing

The TOE intercepts traffic authorized for inspection from monitored clients requesting a serveronly authenticated TLS session with a requested server. The TOE initiates a TLS session with the requested server and validates the requested server’s certificate as legitimately issued by a trusted element of its trust store. The TOE authenticates the server on behalf of the client and generates a certificate that indicates the TOE is an authorized proxy for the requested server. The certificate is issued by the TOE’s embedded CA, which is trusted by the monitored client. The TOE establishes a valid TLS session with the monitored client using the issued certificate. Any TLS traffic between the monitored client and the requested server is decrypted by the TOE and assigned to a unique TLS session thread that is routed to one or more inspection processes, and encrypted into the respective TLS sessions.

2 Conformance Claims

Conformance Statement

This PP-Module inherits exact conformance as required from the specified Base-PPs and as defined in the CC and CEM addenda for Exact Conformance, Selection-Based SFRs, and Optional SFRs (dated May 2017).

No PPs or PP-Modules are allowed to be specified in a PP-Configuration with this PP-Module aside from its supported Base-PP.

CC Conformance Claims

This PP-Module is conformant to Parts 2 (extended) and 3 (conformant) of Common Criteria Version 3.1, Revision 5 [CC].

PP Claim

This PP-Module does not claim conformance to any Protection Profile.

Package Claim

This PP-Module does not claim conformance to any packages.

3 Security Problem Description

The security problem is described in terms of the threats that the TOE is expected to address, assumptions about its operational environment, and any organizational security policies that the TOE is expected to enforce.

A STIP is a network device that embeds limited CA functionality to support the replacement of end-to-end TLS sessions with TLS session threads, making the underlying plaintext available to additional network security functionality. As such, it exposes data within the TOE boundary, and to external processes, which would normally be encrypted. It manages a CA signing key that is trusted by the monitored clients to issue TLS server certificates representing the requested servers for which inspection is authorized.

The proper installation and configuration of the VPN client is critical to its correct operation such that proper handling of the TOE by an administrator is also addressed.

Note that as a PP-Module, all threats, assumptions, and OSPs defined in the Base-PP will also apply to a TOE unless otherwise specified, depending on which of the Base-PPs it extends. The Security Functional Requirements defined in this PP-Module will mitigate the threats that are defined in the PP-Module but may also mitigate some threats defined in the Base-PPs in more comprehensive detail due to the specific capabilities provided by a VPN client.

3.1 Threats

The following threats defined in this PP-Module extend the threats defined by the Base-PP.

T.UNTRUSTED_COMMUNICATION

Untrusted intermediate systems have access to provide unauthorized communications to the TOE, or to manipulate authorized TLS messages in an attempt to compromise the TOE, the monitored clients, or the requested servers. Within this PP-Module, the focus is on an adversary that controls or exploits a requested server that may attempt to cause the device to inappropriately bypass inspection.

Use of weak cryptography can allow adversary access to plaintext intended by the monitored clients to be encrypted. Such access could disclose user passwords that facilitate additional activities against users of monitored clients. Within this PP-Module, the focus is on the use of weak cryptography and adversary attempts to degrade the cryptographic operations within the TLS protocol.

External network security devices may communicate with the TOE to apply security services to the exposed plaintext. An adversary may attempt to gain access the plaintext via misrouting of traffic or manipulate the traffic in such a way as to cause unauthorized exposure, denial of service, or corruption of the underlying plaintext.

T.AUDIT

Certificates issued by the device are trusted by monitored clients, and are required for analysis if traffic processed by the device causes the client to fail or become compromised. Unknown activity related to the issuance and use of certificates can allow an adversary to mask client exploits through or via the TOE, especially if the device fails before the incident can be understood. Unknown activity associated to routing configurations, communications with the TOE, as well as the decision to bypass inspection of traffic can allow an adversary to mask attempts to access monitored clients.

T.UNAUTHORIZED_USERS

In addition to managing administrative credentials, authorized users may have role restrictions to limit their access to the device’s certification authority functionality. In addition to the threat of disclosure or modification of authorized user credentials to users without authorized access to the device, a user with limited access might attempt to extend their access by gaining access to other user’s credentials.

T.CREDENTIALS

In addition to device credentials used in protected communications, the device maintains a trusted certification authority signing key. Any disclosure or unauthorized manipulation of the signing key can result in unintended certificates, signed executable, or signed data that would be trusted by monitored clients. Any modification of the signing key can result in denial of service to inspection capabilities, or to the monitored clients.

T.SERVICES

Manipulation of the device can result in issued certificates being used for unauthorized purposes or abuse of inspection services. An authorized user (AU) (or adversary able to gain access to AU credentials) can access or misuse device services, or disclose sensitive or security critical data.

T.DEVICE_FAILURE

Failure of the certification authority component can result in unauthorized or improperly constrained certificates, or the inability to properly manage the validity of issued certificates. Failure of routing traffic to inspection processing (internal or external) can result in unauthorized disclosure or modification of traffic, or denial of service to monitored clients.

T.UNAUTHORIZED_DISCLOSURE

In addition to general threats to network devices, the TOE controls access to sensitive data that is intended by the monitored client to be encrypted.

T.INAPPROPRIATE_ACCESS

Decryption services applied to traffic between monitored clients and unintended servers can violate privacy laws, or disclose unauthorized traffic to inspection processes. Certification authority signature applied to unauthorized data could facilitate adversary exploits of monitored clients.

3.2 Assumptions

These assumptions are made on the Operational Environment in order to be able to ensure that the security functionality specified in the PP-Module can be provided by the TOE. If the TOE is placed in an Operational Environment that does not meet these assumptions, the TOE may no longer be able to provide all of its security functionality.

All assumptions for the operational environment of the Base-PP also apply to this PP-Module.

A.LIMITED_FUNCTIONALITY is still operative, but the assumed functionality of the TOE includes the behavior needed to satisfy the functional claims of this PP-Module.

A.NO_THRU_TRAFFIC_PROTECTION is still operative, but only for the interfaces in the TOE that are defined by the Base-PP and not the PP-Module.

A.TRUSTED_ADMINISTRATOR is still operative, but the functional claims of this PP-Module offer a limited ability to protect against malicious administrators, which is not within the scope of the original assumption.

A.RESIDUAL_INFORMATION is still operative, but the residual information is expanded to include information relevant to STIP operation (e.g. decrypted SSL/TLS payload, ephemeral keys). This document does not define any additional assumptions.

3.3 Organizational Security Policies

An organization deploying the TOE is expected to satisfy the organizational security policy listed below in addition to all organizational security policies defined by the claimed base PP.

P.AUTHORIZATION_TO_INSPECT

The authority to inspect client traffic may be limited by law, regulation, or policies based on the monitored client, requested server, or nature of the traffic.

4 Security Objectives

4.1 Security Objectives for the TOE

O.AUDIT_LOSS_RESPONSE
The TOE will respond to possible loss of audit records when an audit trail cannot be written to by restricting auditable events.

Addressed by: FAU_STG.4

O.AUDIT_PROTECTION
The TOE will protect audit records against unauthorized access, modification, or deletion to ensure accountability of user actions.

Addressed by: FAU_STG.1 (from Base-PP), FAU_SAR.1 (optional)

O.CERTIFICATES
The TSF must ensure that certificates, certificate revocation lists, and certificate status information are valid.

Addressed by: FIA_X509_EXT.1/Rev (from Base-PP), FIA_X509_EXT.3 (from Base-PP), FDP_CER_EXT.1, FDP_CER_EXT.2, FDP_CER_EXT.3, FDP_CSIR_EXT.1, FIA_ENR_EXT.1, FIA_X509_EXT.1/STIP, FIA_X509_EXT.2/STIP, FDP_PIN_EXT.1 (optional), FIA_ESTC_EXT.2 (optional), FDP_CER_EXT.4 (selection-based), FDP_CER_EXT.5 (selection-based), FDP_CRL_EXT.1 (selection-based), FDP_CSI_EXT.1 (selection-based), FDP_CSI_EXT.2 (selection-based), FDP_OCSP_EXT.1 (selection-based), FDP_OCSPS_EXT.1 (selection-based), FIA_ESTC_EXT.1 (selection-based)

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

Addressed by: FTA_TAB.1 (from Base-PP), FTA_TAB.1/TLS (selection-based)

O.PERSISTENT_KEY_PROTECTION
The TOE will provide appropriate confidentiality and access protection to persistent keys and security critical parameters stored by the TOE.

Addressed by: FCS_STG_EXT.1, FDP_STG_EXT.1, FPT_KST_EXT.1, FPT_KST_EXT.2, FCS_CKM_EXT.5 (selection-based)

O.PROTECTED_COMMUNICATIONS
The TOE will provide protected communication channels for administrators, other parts of a distributed TOE, and authorized IT entities. The TOE will protect data assets when they are being transmitted to and from the TOE, including through intervening untrusted components.

Addressed by: FCS_CKM.4 (from Base-PP), FCS_TLSC_EXT.1 (from Base-PP), FCS_TLSS_EXT.1 (from Base-PP), FTP_ITC.1 (refined from Base-PP), FCS_COP.1/STIP, FCS_TTTC_EXT.1, FCS_TTTC_EXT.5, FCS_TTTS_EXT.1, FDP_PPP_EXT.1, FDP_PRC_EXT.1, FDP_STIP_EXT.1, FDP_TEP_EXT.1, FCS_TTTC_EXT.3 (selection-based), FCS_TTTC_EXT.4 (selection-based), FCS_TTTS_EXT.3 (selection-based), FCS_TTTS_EXT.4 (selection-based), FDP_STIP_EXT.2 (selection-based)

O.RECOVERY
The TOE will have the ability to store and recover to a previous state at the direction of the administrator (e.g., provide support for archival and recovery capabilities).

Addressed by: FPT_FLS.1, FPT_RCV.1

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

Addressed by: FDP_RIP.1

O.SYSTEM_MONITORING
The TOE will provide the ability to generate audit data and send that data to an external IT entity. The TOE will record in audit records: date and time of action and the entity responsible for the action. The TOE will provide the ability to store and review certificate information.

Addressed by: FAU_STG_EXT.1 (from Base-PP), FAU_GEN.1/STIP, FAU_GCR_EXT.1, FAU_SAR.3 (optional), FAU_SCR_EXT.1 (selection-based)

O.TOE_ADMINISTRATION
The TOE will provide mechanisms to ensure that only privileged users are able to log in and configure the TOE, and provide protections for logged-in users. The TOE will ensure that administrative responsibilities are separated across different roles in order to mitigate the impact of improper administrative activities or unauthorized administrative access.

Addressed by: FMT_MOF.1, FMT_SMF.1/STIP, FMT_SMR.2/STIP

4.2 Security Objectives for the Operational Environment

The Operational Environment of the TOE implements technical and procedural measures to assist the TOE in correctly providing its security functionality (which is defined by the security objectives for the TOE). The security objectives for the Operational Environment consist of a set of statements describing the goals that the Operational Environment should achieve. This section defines the security objectives that are to be addressed by the IT domain or by non-technical or procedural means. The assumptions identified in Section 3 are incorporated as security objectives for the environment. The Operational Environment of the TOE implements technical and procedural measures to assist the TOE in correctly providing its security functionality (which is defined by the security objectives for the TOE). This section defines the security objectives that are to be addressed by the IT domain or by nontechnical or procedural means. As indicated above, if requirements supporting an objective on the TOE (in the previous table) are implemented in whole or in part by the platform, the ST should indicate this by an entry in this table with that objective.

All security objectives for the operational environment of the Base-PP also apply to this PP-Module.

OE.NO_THRU_TRAFFIC_PROTECTION is still operative, but only for the interfaces in the TOE that are defined by the Base-PP and not the PP-Module.

OE.RESIDUAL_INFORMATION is still operative, but the residual information is expanded to include information relevant to STIP operation (e.g. decrypted SSL/TLS payload, ephemeral keys).

OE.AUDIT

The operational environment includes an audit server with adequate storage to retain the audit record, and the audit server provides adequate availability, integrity, and access control to the audit record to support operational requirements. Administration of the audit server is separate from that of the SSL/TLS inspection proxy, and can support all required role separations.

Rationale: Each certificate issued by the SSL/TLS inspection device is trusted by monitored clients for the validity period asserted in the certificate. Inappropriate issuance or use of certificates issued by the SSL/TLS inspection device may require the circumstances of the issuance to be investigated, and appropriate actions (e.g., certificate revocation, administrative actions, etc.) to be taken.

OE.CERT_REPOSITORY

The OE provides a certificate repository for storage of certificates (and optionally CRLs) issued by the TSF.

Rationale: Each certificate issued by the SSL/TLS inspection device is trusted by monitored clients for the validity period asserted in the certificate. Inappropriate issuance of certificates, especially certificates with code-signing or which can act as subordinate CAs to issue additional certificates, or inappropriate use of certificates issued by the SSL/TLS inspection device to conduct unauthorized inspection, or to gain access to protected resources may require the circumstances of the issuance to be investigated, and appropriate actions (e.g., certificate revocation, administrative actions, etc.) to be taken.

OE.CERT_REPOSITORY_SEARCH

The OE provides the ability to search a certificate repository for specific certificate fields in certificates issued by the TSF and return the certificate and an identifier for the certificate that can be used to search the audit trail for events related to that certificate and for unauthorized or improperly constrained certificates.

Rationale: Each certificate issued by the SSL/TLS inspection device is trusted by monitored clients for the validity period asserted in the certificate. Inappropriate issuance or use of certificates issued by the SSL/TLS inspection device may require the circumstances of the issuance to be investigated, and appropriate actions (e.g., certificate revocation, administrative actions, etc.) to be taken.

4.3 Security Objectives Rationale

This section describes how the assumptions, threats, and organizational security policies map to the security objectives.

Table 1: Security Objectives Rationale

Threat, Assumption, or OSP	Security Objectives	Rationale
T.UNTRUSTED_COMMUNICATION	O.PROTECTED_COMMUNICATIONS	Data traversing the TOE is subject to authenticity and integrity verification.
T.AUDIT	O.AUDIT_LOSS_RESPONSE	The TOE provides mechanisms to deal with audit trails being unavailable.
	O.AUDIT_PROTECTION	Audit records are protected from modification, deletion, and unauthorized access.
	O.SYSTEM_MONITORING	Audit records contain the

		information necessary to determine cause for concerns.
	OE.AUDIT	Storage within an external audit server provides increased record capacity.
	OE.CERT_REPOSITORY	The certificate repository provides a comprehensive set of certificates generated by the TOE that can be searched.
	OE.CERT_REPOSITORY_SEARCH	Ability to search the audit trail for certificate related events provides confidence in certificate validity and proper use.
T.UNAUTHORIZED_USERS	O.TOE_ADMINISTRATION	Use of role separation and authentication mechanisms ensure that only authorized users can access the TOE.
T.CREDENTIALS	O.CERTIFICATES	The TOE tracks certificates, certificate revocation lists, and certificate status information used by the TSF.
	O.PERSISTENT_KEY_PROTECTION	Keys stored on the TOE are protected from unauthorized use and disclosure.
	OE.CERT_REPOSITORY	A certificate repository for all certificates issued by the TOE is provided, making verification straightforward.
T.SERVICES	O.CERTIFICATES	The TOE verifies certificates, certificate revocation lists, and certificate status information prior to any use.
	O.PROTECTED_COMMUNICATIONS	Data traversing the TOE is subject to authenticity, confidentiality, and integrity verification.
	O.TOE_ADMINISTRATION	Use of role separation and authentication mechanisms mitigates the risk of misuse and improper disclosure.
T.DEVICE_FAILURE	O.CERTIFICATES	The TOE verifies certificates, certificate revocation lists, and certificate status information is valid.
	O.INTEGRITY_PROTECTION	Software, TSF, and user data are protected via integrity mechanisms.
	O.PERSISTENT_KEY_PROTECTION	Keys stored on the TOE are protected from unauthorized use and

		disclosure.
	O.RECOVERY	Administrators have the ability to restore the TOE to a previous (known-good) state.
T.UNAUTHORIZED_DISCLOSURE	O.PROTECTED_COMMUNICATIONS	Data traversing the TOE is subject to authenticity, confidentiality, and integrity verification.
	O.TOE_ADMINISTRATION	Use of role separation and authentication mechanisms mitigates the risk of misuse and ensures the device is properly managed.
T.INAPPROPRIATE_ACCESS	O.RESIDUAL_INFORMATION_CLEARING	The TOE's lack of residual data retention ensures that unauthorized access to information is not possible.
	O.TOE_ADMINISTRATION	Use of role separation and authentication mechanisms mitigates the risk of misuse and improper disclosure.
	OE.RESIDUAL_INFORMATION	Sensitive information residing within the operational environment, such as keys and decrypted data, are unavailable.
P.AUTHORIZATION_TO_INSPECT	O.DISPLAY_BANNER	The TOEs advisory warning includes consent to monitor.
	O.PROTECTED_COMMUNICATIONS	The TSF ensures that data traversing the TOE boundary is protected, alleviating concerns about inspection.
	O.TOE_ADMINISTARTION	Administrator roles provide separation of activities and ensure inspection is authorized and performed properly.

5 Security Requirements

This chapter describes the security requirements which have to be fulfilled by the product under evaluation. Those requirements comprise functional components from Part 2 and assurance components from Part 3 of [CC]. The following conventions are used for the completion of operations:

- **Refinement** operation (denoted by **bold text** or ~~strikethrough text~~): is used to add details to a requirement (including replacing an assignment with a more restrictive selection) or to remove part of the requirement that is made irrelevant through the completion of another operation, and thus further restricts a requirement.
- **Selection** (denoted by *italicized text*): is used to select one or more options provided by the [CC] in stating a requirement.
- **Assignment** operation (denoted by *italicized text*): is used to assign a specific value to an unspecified parameter, such as the length of a password. Showing the value in square brackets indicates assignment.
- **Iteration** operation: is indicated by appending the SFR name with a slash and unique identifier suggesting the purpose of the operation, e.g. "/EXAMPLE1."

5.1 General Purpose Operating Systems PP Security Functional Requirements Direction

In a PP-Configuration that includes the NDcPP, the STIP is expected to rely on some of the security functions implemented by the network device as a whole and evaluated against the Base-PP. In this case, the following sections describe any modifications that the ST author must make to the SFRs defined in the Base-PP in addition to what is mandated by section 5.2.

5.1.1 Modified SFRs

The SFRs listed in this section are defined in the General Purpose Operating Systems PP and relevant to the secure operation of the TOE.

5.1.1.1 Security Audit (FAU)

FAU_STG.1 Protected Audit Trail Storage

FAU_STG.1.1

This PP-Module does not modify this SFR as it is defined in the Base-PP. This SFR is optional in the Base-PP but is mandatory for a TOE that conforms to this PP-Module.

Evaluation Activities

FAU_STG.1
There is no change to the Base-PP EAs for this SFR when this PP-Module is claimed.

5.1.1.2 Cryptographic Support (FCS)

FCS_CKM.4 Cryptographic Key Destruction

FCS_CKM.4.1

The TSF shall destroy all cryptographic keys **and critical security parameters, when no longer required** in accordance with the specified cryptographic key destruction method [**selection**:

- *For plaintext keys in volatile storage, the destruction shall be executed by a [**selection**:*
  - *Single overwrite consisting of [**selection**:*
    - *a pseudo-random pattern using the TSF's RBG,*
    - *zeroes,*
    - *ones,*
    - *a new value of the key,*
    - *[**assignment**: a static or dynamic value that does not contain any CSP]*

    ],
  - *Destruction of reference to the key directly followed by a request for garbage collection*

  ],
- *For plaintext keys in non-volatile storage, the destruction shall be executed the invocation of an interface provided by a part of the TSF that [**selection**:*
  - *Logically addresses the storage location of the key and performs a [**selection**: single, [**assignment**: number of passes]-pass] overwrite*

*consisting of [**selection**:*

- *a pseudo-random pattern using the TSF's RBG,*
- *zeroes,*
- *ones,*
- *a new value of the key,*
- *[**assignment**: a static or dynamic value that does not contain any CSP]*

],

- *Instructs a part of the TSF to destroy the abstraction that represents the key*

]

] that meets the following: [*no standard*].

Application Note: This SFR is refined from its definition in the Base-PP through the inclusion of security critical parameters and clarifies when destruction is required; a STIP device includes persistent keys, including the embedded CA's signing private key that should not be destroyed until they are no longer needed. Security critical parameters includes security related information (e.g., secret and private cryptographic keys, authentication data such as passwords and PINs) appearing in plaintext or otherwise unprotected form and whose disclosure or modification can compromise the security of a CA or the security of the information protected by the CA or the security of the information protected by the CA.

Evaluation Activities

FCS_CKM.4
This SFR is refined in this PP-Module to include requirementsfor destruction of security critical parameters as well as keys. The EA for the Base-PP are extended to include security critical parameters whenever keys are indicated.

FCS_TLSC_EXT.1 TLS Client Protocol Without Mutual Authentication

FCS_TLSC_EXT.1.1

This PP-Module does not modify this SFR as it is defined in the Base-PP. This SFR is selection-based in the Base-PP but is mandatory for a TOE that conforms to this PP-Module because of this PP-Module's modifications to FTP_ITC.1.

Evaluation Activities

FCS_TLSC_EXT.1
There is no change to the Base-PP EAs for this SFR when this PP-Module is claimed.

FCS_TLSS_EXT.1 TLS Server Protocol Without Mutual Authentication

FCS_TLSS_EXT.1.1

This PP-Module does not modify this SFR as it is defined in the Base-PP. This SFR is selection-based in the Base-PP but is mandatory for a TOE that conforms to this PP-Module because of this PP-Module's modifications to FTP_ITC.1.

Evaluation Activities

FCS_TLSS_EXT.1
There is no change to the Base-PP EAs for this SFR when this PP-Module is claimed.

5.1.1.3 Identification and Authentication (FIA)

FIA_X509_EXT.1/Rev X.509 Certificate Validation

FIA_X509_EXT.1.1/Rev

This PP-Module does not modify this SFR as it is defined in the Base-PP. This SFR is selection-based in the Base-PP but is mandatory for a TOE that conforms to this PP-Module because of this PP-Module's modifications to FTP_ITC.1.

Application Note: FIA_X509_EXT.1/STIP defines the TOE's X.509 validation behavior for TLS certificates presented to the TSF as part of TLS proxying. At minimum, FIA_X509_EXT.1/Rev is used by the TOE to validate any certificates

loaded onto it. If the TOE has other functions that require the use of X.509 certificates (e.g. code signing for integrity testing or software updates, TLS interfaces used for a purpose other than session proxying such as audit server or authentication server connections), FIA_X509_EXT.1/Rev applies to those as well.

Evaluation Activities

FIA_X509_EXT.1/Rev
There is no change to the Base-PP EAs for this SFR when this PP-Module is claimed.

FIA_X509_EXT.2 X.509 Certificate Authentication

FIA_X509_EXT.2.1

The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for **TLS,** [**selection**: *DTLS*, *HTTPS*, *IPsec*, *SSH*, ***no other protocols***] and [**selection**: *code signing for system software updates*, *code signing for integrity verification*, *[**assignment**: other uses]*, *no additional uses*].

FIA_X509_EXT.2.2

When the TSF cannot establish a connection to determine the revocation status of a certificate, the TSF shall [**selection**:

- *allow the **[selection: Security Administrator, CA Operations Staff]** to choose whether to **[selection: accept the certificate, associate the failed connection event per FDP_TEP_EXT.1.5]** in these cases,*
- *accept the certificate,*
- *not accept the certificate*

].

Application Note: “TLS” is moved outside the selection in the first element, since the TOE must implement TLS to accomplish the STIP functionality. The application notes for the first element from the Base-PP also apply.

It is worth noting that since this SFR applies to all uses of certificates in the TOE, it may be the case that the actions taken in response to a failure to be able to determine revocation status (which is specified in the 2nd element) is handled differently for different connections. If this is the case, the ST author must make it clear which actions are associated with which connections so that the correct evaluation of the functionality can be performed.

The second element has three modifications from that in the Base-PP. First, the word “validity” is replaced with “revocation status” for clarity. This is consistent with what is in the application note in the NDcPP, and using “revocation status” more directly indicates what is required.

Second, the general notion of “administrator” is replaced with the more refined roles defined in this PP-Module; the ST author should make the appropriate selection.

Finally, a selection is added that allows ST author flexibility in addressing the issue of failure to connect to check revocation status in the specific case that the certificates being checked are associated with either a monitored client or a requested server. This selection (“to associate the failed connection event per FDP_TEP_EXT.1.5”), when chosen, indicates that selected administrative role is able to specify a STIP operation (block, bypass, inspect) to be taken in the event that the revocation status can’t be checked. The requirement that the TOE be able to perform this operation when such an event occurs is specified in FDP_TEP_EXT.1.5.

Evaluation Activities

FIA_X509_EXT.2
There is no change to the Base-PP EAs for this SFR when this PP-Module is claimed.

FIA_X509_EXT.3 X.509 Certificate Requests

FIA_X509_EXT.3.1

In the Base-PP, this SFR is optional but must be claimed in any situation where the TOE presents its own X.509 certificate to an external entity (e.g. any case where the TOE acts as a TLS server or where the TOE acts as a TLS client in an

connection that uses mutual authentication). A STIP TOE must present an X.509 certificate to an external entity as part of TLS session proxying. The TOE may obtain this certificate either using PKCS#10 (covered by this SFR) or through Enrollment over Secure Transport (EST), which is covered by the selection-based SFR FIA_ESTC_EXT.1. Therefore, the ST author only claims FIA_X509_EXT.3 if PKCS#10 is selected in FIA_ENR_EXT.1.

Evaluation Activities

FIA_X509_EXT.3
There is no change to the Base-PP EAs for this SFR when this PP-Module is claimed.

5.1.1.4 Trusted Path/Channels (FTP)

FTP_ITC.1 Inter-TSF Trusted Channel

FTP_ITC.1.1

The TSF shall be capable of using **TLS** and [**selection**: *IPsec*, *SSH*, *DTLS*, *HTTPS*, ***no other protocols***] to provide a trusted communication channel between itself and authorized IT entities supporting the following capabilities: audit server, **TLS session proxying**, [**selection**: *authentication server*, *[**assignment**: other capabilities]*, *no other capabilities*] that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from disclosure and detection of modification of the channel data.

FTP_ITC.1.2

The TSF shall permit the TSF or the authorized IT entities to initiate communication via the trusted channel.

FTP_ITC.1.3

The TSF shall initiate communication via the trusted channel for **establishment of TLS proxy connections,** [**assignment**: *list of services for which the TSF is able to initiate communications*].

Evaluation Activities

FTP_ITC.1
There is no change to the Base-PP EAs for this SFR when this PP-Module is claimed.

5.2 TOE Security Functional Requirements

The following section describes the SFRs that must be satisfied by any TOE that claims conformance to this PP-Module. These SFRs must be claimed regardless of which PP-Configuration is used to define the TOE.

5.2.1 Auditable Events for Mandatory SFRs

Table 2: Auditable Events for Mandatory Requirements

Requirement	Auditable Events	Additional Audit Record Contents

5.2.2 Security Audit (FAU)

FAU_GCR_EXT.1 Generation of Certificate Repository

FAU_GCR_EXT.1.1

The TSF shall [**selection**: *store*, *invoke the Operational Environment to store*] certificates issued by the TSF.

Application Note: While there is a requirement that a certificate repository exists and the TOE stores all certificates it generates in that repository, the repository can physically be within the TOE or in the OE. If the repository is provided by the TOE, then the first item in the first selection is chosen. If the storage is provided by the OE, then the second item in the first selection is chosen. It should be noted that the physical implementation of the certificate repository is left to the vendor; for instance, it can be a standalone store, or incorporated within the audit trail.

Evaluation Activities

FAU_GCR_EXT.1
TSS
The evaluator shall examine the TSS to determine that it describes the certificate repository. If the certificate repository is provided by the OE, the evaluator shall check the TSS to ensure it describes the interfaces invoked by the TOE to store certificates.
Guidance
The evaluator shall ensure that the guidance describes any operations necessary to cause certificates to be stored in the repository.
Tests
The evaluator shall cause a certificate to be generated by the TSF. The evaluator shall confirm that the certificate is stored in the certificate repository.

FAU_GEN.1/STIP Audit Data Generation (STIP)

FAU_GEN.1.1/STIP

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 [*not specified*] level of audit; and
c. [*auditable events defined in Auditable Events table*]

Application Note: The "Start-up and shtudown of the audit functions" event is identical to the event defined in the Base-PP's iteration of FAU_GEN.1. The TOE is not required to have two separate events for this behavior if there is only a single audit stream that which all audit events use. If the TOE does maintain a separate logging facility for STIP-related behavior, then this event must be addressed for it. Note that if the audit functions cannot be started and stopped separately from the TOE itself, then auditing the start-up and shutdown of the TOE is sufficient to address this.

FAU_GEN.1.2/STIP

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

a. Date and time of the event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event; and
b. For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [*additional information defined in Auditable Events table for each auditable event, where applicable*].

TLS session thread identifier, identifier(s) of processing element(s) bypassed, reason for bypass

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GCR_EXT.1	None	
FAU_SCR_EXT.1 (selection-based)	None	
FAU_SAR.1 (optional)	None	
FAU_SAR.3 (optional)	None	
FAU_STG.4 (optional)	None	
FCS_CKM_EXT.5 (selection-based)	None	
FCS_STG_EXT.1	None	
FCS_TTTC_EXT.1	Establishment of TLS session	TLS session parameters
FCS_TTTC_EXT.3 (selection-based)	Mutual authentication authorized	[**selection**: *client certificate value*, [***assignment**: client certificate object identifier*]]
	Mutual authentication not authorized	None

FCS_TTTC_EXT.4 (selection-based)	None	
FCS_TTTC_EXT.5	None	
FCS_TTTS_EXT.1	Establishment of TLS session	TLS session parameters
FCS_TTTS_EXT.3 (selection-based)	Mutual authentication required and valid client certificate received	Client certificate
	Mutual authentication not used	None
FCS_TTTS_EXT.4 (selection-based)	None	
FDP_CER_EXT.1	None	
FDP_CER_EXT.2	Linking of issued certificate to validated certificate	Success: [**selection**: *issued certificate value, issued certificate object identifier*], [**selection**: *validated certificate value, validated certificate object identifier*] Failure: Reason for failure
FDP_CER_EXT.3	Certificate generation	Success: [**selection**: *certificate value, certificate object identifier*]
FDP_CER_EXT.4 (selection-based)	None	
FDP_CER_EXT.5 (selection-based)	Certificate generation	Success: [**selection**: *certificate value, certificate object identifier*]
FDP_CRL_EXT.1 (selection-based)	Failure to generate CRL	None
FDP_CRL_EXT.2 (selection-based)	None	
FDP_CSI_EXT.1 (selection-based)	None	
FDP_CSI_EXT.2 (selection-based)	None	
FDP_CSIR_EXT.1 (selection-based)	None	
FDP_OCSP_EXT.1 (selection-based)	Failure to generate certificate status information	None
FDP_OCSPS_EXT.1 (selection-based)	Failure to include certificate status information in TLS handshake message	None
FDP_PIN_EXT.1 (optional)	None	
FDP_PPP_EXT.1	Configuration changes to the plaintext processing policy	None
FDP_PRC_EXT.1	Plaintext routed to inspection processing functional component	TLS session thread identifier, [**assignment**: *processing element*

		identifier]
FDP_RIP.1	None	
FDP_STG_EXT.1	None	
FDP_STIP_EXT.1	Establishment of a TLS inspection session thread	[**assignment**: *TLS session thread attributes*], [**assignment**: *client attributes*], and [**assignment**: *server attributes*] associated with the thread
	Establishment of an encrypted TLS data flow	[**assignment**: *encrypted TLS data flow attributes*]
	Bypass operation invoked	
	Block operation invoked	TLS session thread identifier, reason for blocking
FDP_STIP_EXT.2 (selection-based)	None	
FDP_TEP_EXT.1	Mutual authentication authorized	[**assignment**: *client attributes obtained from valid client certificate*]
FIA_ENR_EXT.1	None	
FIA_ESTC_EXT.1 (selection-based)	EST requests	None
FIA_ESTC_EXT.2 (objective)	None	
FIA_X509_EXT.1/STIP	None	
FMT_MOF.1	None	
FMT_SMF.1/STIP	None	
FMT_SMR.2/STIP	None	
FPT_FLS.1	Invocation of failures under this requirement	Indication that the TSF has failed with the type of failure that occurred
FPT_KST_EXT.1	None	
FPT_KST_EXT.2	All attempts to use the TOE's embedded CA's private signing key and [**selection**: *[**assignment**: other secret and private keys], no other secret and private keys*]	Identifier of user or process that attempted access
FPT_RCV.1	The fact that a failure or service discontinuity occurred	None
	Resumption of regular opartion	TSF failure types that are available on recovery
FTA_TAB.1/TLS (selection-based)	None	

Table 3: Auditable Events

Application Note: The ST author only needs to include the rows from the Auditable Events table that correspond to the SFRs claimed in the ST. The TOE is not required to generate auditable events for selection-based or optional SFRs that it does not claim.

Evaluation Activities

FAU_GEN.1/STIP
TSS
The evaluator shall examine the TSS to verify that it describes the audit mechanism(s) that the TOE uses to generate audit records for STIP behavior. If any audit mechanisms the TSF uses for this are not used to generate audit records for events defined by FAU_GEN.1 in the Base-PP, the evaluator shall ensure that any STIP-specific audit mechanisms also meet the relevant functional claims from the Base-PP.

For example, FAU_STG_EXT.1 requires all audit records to be transmitted to the OE over a trusted channel. This includes the audit records that are required by FAU_GEN.1/STIP. Therefore, if the TOE has an audit mechanism that is only used for STIP functionality, the evaluator shall ensure that the STIP related audit records meet this requirement, even if the mechanism used to generate these audit records does not apply to any of the auditable events defined in the Base-PP.
Guidance
The evaluator shall examine the operational guidance to verify that it identifies all security-relevant auditable events claimed in the ST and includes sample records of each event type. If the TOE uses multiple audit mechanisms to generate different sets of records, the evaluator shall verify that the operational guidance identifies the audit records that are associated with each of the mechanisms such that the source of each audit record type is clear.
Tests
The evaluator shall test the audit functionality by performing actions that trigger each of the claimed audit events and verifying that the audit records are accurate and that their format is consistent with what is specified in the operational guidance. The evaluator may generate these audit events as a consequence of performing other tests that would cause these events to be generated.

FAU_STG.4 Prevention of Audit Data Loss

FAU_STG.4.1

The TSF shall [*prevent audited events, except those taken by the **[assignment: Security Administrator, Auditor]***] and [**assignment**: *other actions to be taken in case of audit storage failure*] if the audit trail **cannot be written to**.

Application Note: This requirement applies to the TOE regardless of whether the audit trail is stored within the TOE boundary or on an external system in the Operational Environment. If the audit trail is stored locally, then the requirement applies when the audit trail cannot be written to when it is full. If the audit trail (in whole or in part) is stored on a system external to the TOE, then the requirement applies when the connection between the TOE and the external audit server becomes disconnected and the audit trail cannot be written to. In the case where the audit trail is external to the TOE and cannot be written to because it is full (and the TOE has some way of detecting that), then the requirement applies in that case as well. In all cases, the ST author is expected to describe (in the TSS) how the TSF is made aware of any such failures and how it behaves in response.

Evaluation Activities

FAU_STG.4
TSS
The evaluator shall examine the TSS to ensure it describes the behavior of the TSF when the audit trail cannot be written to. The evaluator shall ensure the TSS describes where the audit trail is stored (locally, remotely, or both), how the TSF detects audit full conditions if the audit trail is stored locally, whether and how the TSF detects audit full conditions for remote audit repositories, and how the TSF detects loss of communication with external audit repositories (if using an external audit server). The evaluator shall also ensure the TSS describes what actions can be performed by the privileged user, if any, in each case where the audit trail cannot be written.
Guidance
The evaluator shall examine the operational guidance to ensure it describes what conditions result in the audit trail not being able to be written to, and how an Auditor recognizes that such a condition has occurred. The evaluator shall also examine the operational guidance to ensure it includes remedial steps for correcting these issues.
Tests
The evaluator shall perform the following tests. The tests are conditional on where the audit data are being stored.

Test 1 demonstrates the capability of the TOE to react to an indication that the repository is full; this is always applicable if the audit data are stored locally. If the TOE has a means to detect

that a remote audit repository is full, then this test will be run for those types of TOEs as well. Test 2 is only executed in cases where an external repository is supported, and tests the ability of the TOE to detect when the connection to the repository becomes unavailable:

- ***Test 1:*** *(conditional) The evaluator shall cause the audit trail to become full, verify that the TSF behaves as documented in the TSS, and verify that a privileged user can perform the documented remedial steps.*
- ***Test 2:*** *(conditional) The evaluator shall cause the audit trail to become unavailable, verify that the TSF behaves as documented in the TSS, and verify that a privileged user can perform the documented remedial steps.*

5.2.3 Cryptographic Support (FCS)

FCS_COP.1/STIP Cryptographic Operation (Data Encryption/Decryption in Support of STIP

FCS_COP.1.1/STIP

The TSF shall perform encryption/decryption in accordance with a specified cryptographic algorithms[*AES in CCM and CCM-8 mode and [**selection**: TDES used in CBC mode with 3 distinct keys in its key set, no other algorithms]*] and cryptographic key sizes [**selection**: *128 bits, 192 bits, 256 bits*] that meet the following: [*AES as specified in ISO 18033-3, CCM and CCM-8 as specified in NIST SP 800-38C and [**selection**: TDES as specified in NIST SP 800-67 Rev 2 and CBC mode as specified in NIST SP 800-38A addendum, no other standards]*].

Application Note: This requirement, in conjunction with FCS_COP.1/DataEncryption from the BasePP, is used to support FCS_TTTC_EXT.1 and FCS_TTTS_EXT.1. Note that FCS_TTTC_EXT.1 and FCS_TTTS_EXT.1 may necessitate certain selections.

Evaluation Activities

FCS_COP.1/STIP
TSS
The evaluator shall verify that the TSS includes a description of encryption functions used for user data encryption, and that this description includes the key sizes and modes of operation.

The evaluator shall check that the TSS describes how the TOE satisfies constraints on key sizes specified in the SFR.
Guidance
The evaluator shall verify that the AGD guidance documentation includes instructions for meeting this requirement, including any configuration required to ensure the TSF only supports Triple Data Encryption Standard (TDES) with three distinct keys.
Tests

- ***Test 1:*** *The evaluator shall verify the AES implementation used to support TLS cipher suites in accordance with the requirements by conducting the following tests:*

  AES-CCM Test

  The evaluator shall perform the following tests.

  Preconditions for testing:

  - *Specification of keys as input parameter to the function to be tested*
  - *Specification of required input parameters such as modes*
  - *Specification of user data (plaintext)*
  - *Tapping of encrypted user data (ciphertext) directly in the non-volatile memory*

  These tests are intended to be equivalent to those described in the NIST document, "The CCM Validation System (CCMVS)," updated 9 Jan 2012, found at . It is not recommended that evaluators use values obtained from static sources such as or use values not generated expressly to exercise the AES-CCM implementation.

  The evaluator shall test the generation-encryption and decryption-verification functionality of AES-CCM for the following input parameter and tag lengths:

  - ***Keys:*** *All supported and selected key sizes (e.g., 128, 256 bits).*
  - ***Associated Data:*** *Two or three values for associated data length: The minimum (≥ 0 bytes) and maximum (≤ 32 bytes) supported associated data lengths, and 2^16 (65536) bytes, if supported.*
  - ***Payload:*** *Two values for payload length: The minimum (≥ 0 bytes) and maximum (≤ 32 bytes) supported payload lengths.*
  - ***Nonces:*** *All supported nonce lengths (7, 8, 9, 10, 11, 12, 13) in bytes.*
  - ***Tags:*** *All supported tag lengths (4, 6, 8, 10, 12, 14, 16) in bytes.*

  The testing for CCM consists of five tests. To determine correctness in each of the below tests, the evaluator shall compare the ciphertext with the result of encryption of the same

inputs with a known good implementation.

Variable Assocated Data Test

For each supported key size and associated data length, and any supported payload length, nonce length, and tag length, the evaluator shall supply one key value, one nonce value, and 10 pairs of associated data and payload values, and obtain the resulting ciphertext.

Variable Payload Test

For each supported key size and payload length, and any supported associated data length, nonce length, and tag length, the evaluator shall supply one key value, one nonce value, and 10 pairs of associated data and payload values, and obtain the resulting ciphertext.

Variable Nonce Test

For each supported key size and nonce length, and any supported associated data length, payload length, and tag length, the evaluator shall supply one key value, one nonce value, and 10 pairs of associated data and payload values, and obtain the resulting ciphertext.

Variable Tag Test

For each supported key size and tag length, and any supported associated data length, payload length, and nonce length, the evaluator shall supply one key value, one nonce value, and 10 pairs of associated data and payload values, and obtain the resulting ciphertext.

Decryption-Verification Process Test

To test the decryption-verification functionality of AES-CCM, for each combination of supported associated data length, payload length, nonce length, and tag length, the evaluator shall supply a key value and 15 sets of input plus ciphertext, and obtain the decrypted payload. Ten of the 15 input sets supplied should fail verification and five should pass.

- ***Test 2:*** *(conditional): The evaluator shall test the TDES implementation used to support TLS cipher suites in accordance with NIST SP 800-67 Rev 2, by conducting the following tests:*

  Variable Plaintext/Ciphertext Known Answer Test:

  For i=1..64, the evaluator shall verify the encrypt functionality by using Key1=Key2=Key3=0x0101010101010101, and IV=0x0000000000000000, to encrypt plaintext p_1{i}=ith basis vector input as a type 2 input, and comparing the resulting ciphertext=c_1{i} output as a type 2 output to known results indicated in table A.1 of NIST SP800-20.

  For i=1..64, evaluator shall verify the decrypt functionality by using Key1=Key2=Key3=0x0101010101010101, and IV=0x0000000000000000, to decrypt ciphertext, c_1{i} and verifying the resulting plaintext to p_1{i}, the ith basis vector.

  Inverse/Initial Permutation Known Answer Test: *For i=1..64, the evaluator shall verify the encrypt functionality by using Key1=Key2=Key3=0x0101010101010101, and IV=0x0000000000000000, to encrypt plaintext p_2{i}=c_1{i} from the Variable Plaintext Known Answer Test, input as a type 5 input, and verifying the resulting ciphertext, c_2{i} output as type 2 output, is equal to the ith basis vector, p_1{i}.*

  For i=1..64, the evaluator shall verify the decrypt functionality by using Key1=Key2=Key3=0x0101010101010101, and IV=0x0000000000000000, to decrypt ciphertext c_2{i]=p_1{i}, input as input type 5, and verifying the resulting plaintext, p_2{i} output as type 2 output, is equal to c_1{i}.

  Variable Key Known Answer Test:

  For i=1..64 not zero mod 8, the evaluator shall verify the encrypt function using Key1{i}=Key2{i}=Key3{i} equal to the vector consisting of a one in the ith position, zeros in all other positions not zero mod 8, and parity bits in positions 0 mod 8 computed to make each byte have odd parity, and using IV=0x0000000000000000, to encrypt plaintext p_3{i}=0x0000000000000000, input as a type 2 input, and comparing the resulting ciphertext, c_3{i} output as a type 2 output to known results indicated in table A.2 of NIST SP800-20.

  For i=1..64 not zero mod 8, the evaluator shall verify the decrypt functionality using the same Key1{i}=Key2{i}=Key3{i} above, and IV=0x0000000000000000, to decrypt ciphertext c_3{i} and comparing the resulting plaintext to 0x0000000000000000.

  Permutation Operation Known Answer Test:

  For i=0..31, the evaluator shall verify the encrypt functionality by using Key1{i}=Key2{i}=Key3{i} equal to the round i key in table A.3 of NIST SP800-20, and IV=0x0000000000000000 to encrypt plaintext = 0x0000000000000000, and verifying that the resulting ciphertext c4{i} matches the known result for round I indicated in table A.3 of NSIT SP800-20.

  For i=0..31, the evaluator shall verify the decrypt functionality by using

Key1{i}=Key2{i}=Key3{i} equal to the round I key in table A.3 of NIST SP800-20, and IV=0x0000000000000000 to decrypt ciphertext c4{i} above, and verifying that the resulting plaintext for each round equals 0x0000000000000000.

Substitution Table Known Answer Test

For i=0..18, the evaluator shall verify the encrypt functionality by using Key1{i}=Key2{i}=Key3{i} equal to the round i key in table A.4 of NIST SP800-20, and IV=0x0000000000000000 to encrypt the round i plaintext, p4{i} in table A.4 of NIST SP300-20, and verifying that the resulting ciphertext c4{i} matches the known result for round i indicated in table A.4 of NSIT SP800-20.

For i=0..18, the evaluator shall verify the decrypt functionality by using Key1{i}=Key2{i}=Key3{i} equal to the round i key in table A.4 of NIST SP800-20, and IV=0x0000000000000000 to decrypt ciphertext =c4{i} above, and verifying that the resulting plaintext matches p4{i} above.

Monte Carlo Test:

Three-key test:

- *The evaluator shall conduct the Monte Carlo Test for the Cipher Block Chaining (CBC) mode of Triple Data Encryption Algorithm (TDEA) encryption indicated in NIST SP 800-20 Section 2.1.5.6 against the TOE, using three distinct keys, Key1 not equal to Key2, Key2 not equal to Key3 and Key3 not equal to Key1, and validate the results against a known good implementation of TDEA.*
- *The evaluator shall conduct the Monte Carlo Test for the CBC mode of TDEA decryption indicated in NIST SP 800-20 Section 2.2.5.6 against the TOE, using three distinct keys, Key1 not equal to Key2, Key2 not equal to Key3 and Key3 not equal to Key1, and validate the results against a known good implementation of TDEA.*

FCS_STG_EXT.1 Cryptographic Key Storage

FCS_STG_EXT.1.1

Persistent private and secret keys shall be stored within the TSF using [**assignment**: *method of hardware-protected storage*].

Application Note: This requirement ensures that persistent secret keys and private keys are stored securely when not in use. Methods of hardware protected storage can be direct or via encryption with a KEK which is protected by hardware. The application notes for FPT_KST_EXT.2.1 contain further discussion of private and secret keys referenced by this SFR

Evaluation Activities

FCS_STG_EXT.1
TSS
The evaluator will check the TSS to ensure it lists each persistent secret and private key needed to meet the requirements in the ST. For each of these items, the evaluatorshall confirm that the TSS lists for what purpose it is used, and how it is stored, and that the storage is hardware-protected.
Guidance
There are no guidance EAs for this component.
Tests
There are no test EAs for this component.

FCS_TTTC_EXT.1 Thru-Traffic TLS Inspection Client Protocol

FCS_TTTC_EXT.1.1

The TSF shall implement [*TLS 1.2 (RFC 5246), TLS 1.0 (RFC 2246), and [**selection**: TLS 1.1 (RFC 4346), no other TLS versions]*] as a client to the requested server that supports the following cipher suites: [

- *TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289*
- *TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289*
- *TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289*
- *TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289*
- *TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288*
- *TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246*
- *TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288*
- *TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246*
- *TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289*
- *TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289*
- *TLS_DHE_RSA_ WITH_AES_256_CCM as defined in RFC 6655*

- *TLS_RSA_ WITH_AES_256_CCM as defined in RFC 6655*
- *TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289*
- *TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289*
- *TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289*
- *TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289*
- *TLS_DHE_RSA_WITH_AES_128_CCM as defined in RFC 6655*
- *TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288*
- *TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246*
- *TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289*
- *TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289*
- *TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289*
- *TLS_RSA_WITH_AES_128_CCM as defined in RFC 6655*
- *TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288*
- *TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246*
- *TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 8422*
- *TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 8422*
- *TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 5246*
- *TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 8422*
- *TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 5246*
- *TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 8422*
- *TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 8422*
- *TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 8422*
- *TLS_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246*
- *TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5426*
- *TLS_RSA_WITH_AES_128_CCM_8 as defined in RFC 6655*
- *TLS_DHE_RSA_WITH_AES_128_CCM_8 as defined in RFC 6655*
- *TLS_DHE_RSA_WITH_AES_256_CCM_8 as defined in RFC 6655*
- *TLS_RSA_WITH_AES_256_CCM_8 as defined in RFC 6655*
- *[**selection**: TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA as defined in RFC 8422, TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA as defined in RFC 5246, TLS_RSA_WITH_3DES_EDE_CBC_SHA as defined in RFC 5246, [**assignment**: other supported cipher suites], no other cipher suites]*

] and also supports functionality for [**selection**:

- *mutual authentication*,
- *session renegotiation*,
- *neither mutual authentication nor session renegotiation*

].

Application Note: TLS version 1.2 and 1.0 must be supported; support for TLS version 1.1 is optional, and should be chosen if the STIP supports it. The list of cipher suites to support is mandatory but includes some selections in order to support legacy servers that may be required by the monitored clients; additional cipher suites can be included in the assignment. The order of the cipher suites above should be maintained in the ST; FCS_TTTC_EXT.1.4 indicates that the cipher suites are presented in order of preference in the Client Hello sent to the requested server, and that preference is defined as the order in the above SFR.

The above list (as instantiated in the ST) limits the cipher suites that may be proposed by the TOE to the requested server. Behavior if the requested server responds with a cipher suite that is not in the list is defined in FDP_TEP_EXT.1.8.

The selection should indicate if mutual authentication and/or session renegotiation is supported. These selections must be the same for both FCS_TTTC_EXT.1.1 and FCS_TTTS_EXT.1.1. If mutual authentication is selected, the requirements in Section B.4 will be included by the ST author. For this technology, mutual authentication is not desirable on these connections because the STIP will have to issue a certificate representing the client to the requested server, and the server will have to have a trust anchor for that certificate. If session renegotiation is selection, FCS_TTTC_EXT.4 from Section B.5 will be included by the ST author.

The data encryption and decryption algorithms used in this element are performed in accordance with FCS_COP.1/STIP.

FCS_TTTC_EXT.1.2

The TSF shall verify that the presented identifier matches the reference identifier of the server requested by the monitored client using methods described in RFC 6125 section 6 for DNS name types, and via exact, byte-by-byte matching for IP address name types.

Application Note: The rules for verification of identity are described in Section 6 of RFC 6125. The monitored client may specify the server name in the SNI

extension of the Client Hello, or via some other method (e.g., DNS lookup) supported by the TSF. The method for determining that the identity presented matches that expected by the client should be fully described in the ST.

Additionally, support for use of IP addresses in the Subject Name or Subject Alternative Name of TLS server certificates is discouraged as against best practices but may be implemented by requested servers. When no DNS name type reference ID is available from the monitored client and the certificate presented by the requested server includes an IP address name type, exact byte-by-byte matching of the IP address to an IP address reference ID is required. If the certificate does not contain an identifier of type IP address, and no other name type is included as a reference Identifier, the IP address from the underlying transport layer protocol between the TSF and the requested servers should match the IP address reference identifier.

FCS_TTTC_EXT.1.3

The TSF shall validate the certificate presented by the server and terminate the connection if the certificate is invalid, except as allowed by FIA_X509_EXT.2.2.

Application Note: Validity is determined by the identifier verification, certificate path, the expiration date, processing of critical extensions, and the revocation status in accordance with RFC 5280. Certificate validity is specified by and tested in accordance with FIA_X509_EXT.1/STIP. The result of the checks will be one of 1) the certificate is valid; 2) the certificate is invalid; 3) the validity of the certificate is indeterminate because a connection cannot be established to check the revocation status of the certificate (but all other validity checks have passed). FCS_X509_EXT.2.2 (in conjunction with FDP_TEP_EXT.1.5) indicates what the TSF is supposed to do if a connection cannot be established to check the revocation status for this connection (the TOE to a requested server).

FCS_TTTC_EXT.1.4

The TSF shall formulate the Client Hello such that it presents the highest version of the TLS protocol supported by the proxy function in the version field, and presents the list of cipher suites in descending order of preference associated with requested server.

Application Note: This applies to the initial Client Hello sent to the requested server. This may result in a connection being established for an inspect operation, or may not lead to a connection if a bypass or block operation is determined. It should be noted that this transaction may be made even though the result will eventually be block or bypass, because the rule (see FDP_TEP_EXT.1) may require the verified identity of the server, so this connection would be required so that the server certificate could be obtained and verified.

Evaluation Activities

FCS_TTTC_EXT.1.1
TSS
The evaluator will check the description of this protocol in the TSS to ensure that the TLS versions and cipher suites supported for inspection of TLS sessions are included. The evaluator shall check the TSS to ensure that the TLS versions and cipher suites specified for processing such traffic include those listed in FCS_TTTC_EXT.1.1, and no others. The evaluator shall ensure the TSS describes how the cipher suites included in a Client Hello message to a specific requested server might be restricted in accordance with allowances described in the TLS session establishment policy.
Guidance
The evaluator shall check the guidance documentation to ensure it contains instructions on configuring the TOE so that the versions and cipher suites used conform with FCS_TTTC_EXT.1.1 and the configured TLS session establishment policy.

The evaluator shall verify that the AGD guidance includes instructions for setting the reference identifier to be used for the purposes of certificate validation in TLS.
Tests
The evaluator shall establish one or more monitored clients and requested servers that are configured to pass TLS sessions through the TOE, and configure the SSL/TLS inspection proxy policy to use the inspection operation for these clients and servers with all supported versions and cipher suites in its allowed set. The evaluator shall configure the monitored client to present a TLS Client Hello with TLS version 1.2 and the full list of supported cipher suites, and use the SNI extension to indicate the DNS name of the requested server for each test. The evaluator shall establish a certification authority (the trusted CA) able to issue certificates for the servers as indicated in the following tests, and install the certification authority's certificate in appropriate trust anchors within the TSF to validate the issued certificates. Additional configuration instructions for the monitored client, the requested server or the server's

certificate are indicated in each of the tests:

- ***Test 1:*** *For each version and cipher suite combination supported, as indicated in FCS_TTTC_EXT.1.1, the evaluator shall configure a server requested by a monitored client to negotiate the version and cipher suite, and issue the server a certificate from the trusted CA containing a subjectAltName (SAN) extension containing the expected DNS name of the server, and which is valid in accordance with FIA_X509_EXT.1/Rev. The evaluator shall then initiate a TLS session from a monitored client through the TOE to the requested server, as indicated in the SNI extension of the Client Hello, and observe that the TLS session between the TOE and the requested server cipher suites is successful. Additionally, the evaluator shall verify that the Client Hello sent from the TSF to the requested server contains the full, ordered list of cipher suites supported for the selected version in accordance with FCS_TTTC_EXT.1.4.*
- ***Test 2:*** *The evaluator shall choose a supported version and cipher suite combination. For each extendedKeyUsage condition for server certificates that allows the TLS session to be completed, as indicated in FIA_X509_EXT.1.1/STIP, the evaluator shall configure a requested server to negotiate the version and cipher suite combination and issue the requested server a new certificate from the trusted CA that has the indicated extendedKeyUsage condition, and is otherwise identical to the certificate used for the similarly configured server from Test 1. The evaluator shall configure the server to present the new certificate in its TLS handshake. The evaluator shall then make a TLS request in turn from a monitored client to each of the reconfigured serversthrough the TOE, and observe that the TLS session from the TOE to the requested server is established.*
- ***Test 3:*** *The evaluator shall establish a new certificate for a server as configured for Test 2a where the extendedKeyUsageextended key usage field is present, does not include either the 'Any' purpose or ServerAuthentication purpose and which does contain the CodeSigning purpose, and configure the server to present the new certificate in its TLS handshake. The evaluator shall make a request to that server from a monitored client through the TOE and verify that the TLS session between the TSF and the server is attempted, but fails.*
- ***Test 4:*** *For each of the following, the evaluator shall issue a new certificate as specified from the trusted CA containing the indicated public key type for a server configured to negotiate a supported version and cipher suite as specified, so the server presents a certificate with a signature or static public key type that is incompatible with the negotiated cipher suite:*
  a. *For a supported cipher suite that uses RSA for signature, the evaluator shall issue a certificate containing an Elliptic Curve Digital Signature Algorithm (ECDSA) public key to represent a server configured to negotiate the cipher suite*
  b. *For a supported cipher suite that uses ECDSA for signature, the evaluator shall issue a certificate containing an RSA public key to represent a server configured to negotiate the cipher suite*
  c. *For a supported cipher suite that uses RSA for key transport, the evaluator shall issue a certificate containing a Diffie-Hellman (DH) public key to represent a server configured to negotiate the cipher suite*
  d. *For a supported cipher suite that uses RSA for key transport, the evaluator shall issue a certificate containing an Elliptic-Curve Diffie-Hellman (ECDH) public key to represent a server configured to negotiate the cipher suite*
  e. *For a supported cipher suite that uses static DH key establishment, the evaluator shall issue a certificate containing an RSA public key to represent a server configured to negotiate the cipher suite*
  f. *For a supported cipher suite that uses static DH key establishment, the evaluator shall issue a certificate containing an ECDH public key to represent a server configured to negotiate the cipher suite*
  g. *For a supported cipher suite that uses static ECDH, the evaluator shall issue a certificate containing an RSA public key that represents a server configured to negotiate the cipher suite*
  h. *For a supported cipher suite that uses static ECDH, the evaluator shall issue a certificate containing a DH public key that represents a server configured to negotiate the cipher suite.*

  The evaluator shall make, in turn, a TLS request to each so-configured server from a monitored client. In each case, the evaluator shall observe that the TSF attempts to establish a TLS session with the requested server and after the server negotiates the cipher suite, the evaluator shall send the new certificate in a server certificate message to the TSF in the place of the expected certificate message, and observe that the TSF does not establish a TLS session with the server.
- ***Test 5:*** *The evaluator shall configure a server to select the TLS_NULL_WITH_NULL_NULL cipher suite. The evaluator shall make a request from a monitored client to the so configured server and verify that the TLS session between the TSF and the server is attempted but not established.*
- ***Test 6:*** *For each of the following, the evaluator shall configure a requested server to negotiate a supported version and cipher suite, as indicated, and use a valid certificate from the trusted CA, but send TLS messages as indicated and otherwise respond as a valid TLS server. For each in turn, the evaluator shall initiate a TLS connection between a monitored client and the requested server through the TOE and observe the indicated behavior of the TOE on receiving the server message:*

- *Test 6.1: Configure the requested server to send an undefined TLS version (for example, 1.5 represented by the two bytes 03 06) and verify that the TSF rejects the connection.*
- *Test 6.2: Configure the requested server to send a Server Hello with the TLS version set to SSL 3.0 (represented by the two bytes 03 00) and verify that the TSF rejects the connection.*
- *Test 6.3: Configure the requested server to use a DHE cipher suite and configure the requested server to send a Server Hello message with at least one byte in the server's nonce in the Server Hello handshake message modified from the expected response, and verify that the TSF rejects the connection. Repeat this test using a requested server configured to use an ECDHE cipher suite and observe that the TSF rejects the connection.*
- *Test 6.4: Configure the requested server to respond to a Client Hello with a cipher suite that is not supported by the TSF, and therefore not present in the Client Hello received by the server. The evaluator shall verify that the TSF rejects the connection.*
- *Test 6.5: Using requested servers configured to use a cipher suite using DHE, and send a KeyExchange handshake message with an invalid signature (e.g., by modifying the signature block in the expected KeyExchange handshake message), and verify that the TSF rejects the connection. Repeat this test with a requested server configured to use a cipher suite using ECDHE and verify that the TSF rejects the connection.*
- *Test 6.6: Configure the requested server to respond with an invalid Server Finished message (e.g., by modifying a byte in the expected Server Finished handshake message) and verify that the TSF rejects the connection.*

FCS_TTTC_EXT.1.2

TSS

The evaluator shall ensure that the TSS describes the TSF method of establishing all reference identifiers for through-traffic processing, including which types of reference identifiers are supported and whether IP addresses and wildcards are supported. The evaluator shall ensure that the TSS describes how the TSF determines reference identifiers from the various identity attributes associated to the requested server and match what is expected by the monitored client. The evaluator shall ensure that the TSS describes how the reference identifiers are matched to the identifiers presented in the server's certificate.

Guidance

The evaluator shall ensure that the guidance contains instructions on establishing reference identifiers if supported through an administrative interface.

Tests

Using the setup for FCS_TTTC_EXT.1.1, the evaluator shall perform the following tests. Note that Test 1 of FCS_TTTC_EXT.1.1 confirms the TSF properly validates the reference ID of a certificate containing a DNS name in the subjectAltName matching the SNI contained in the Client Hello of a monitored client, and is not repeated. The remaining tests cover support for other name forms and negative testing.

- ***Test 1:** The evaluator shall issue a certificate from the trusted CA that represents a requested server that contains a SAN extension with a valid DNS name type. The evaluator shall configure the requested server to use a valid, supported version and cipher suite combination consistent with the certificate, and provide the certificate in response to a TLS request. The evaluator shall establish a TLS session from a monitored client to the requested server through the TOE using an SNI extension in the Client Hello that does not match the name in the certificate. The evaluator shall ensure the TOE does not succeed in establishing a TLS connection to the requested server.*
- ***Test 2:** (conditional, the TSF supports additional reference identifiers not used in FCS_TTTC_EXT.1.1 test 1): For each additional reference identifier described in the TSS, the evaluator shall establish a monitored client and requested server that causes the TSF to establish a reference identifier of the indicated type. The evaluator shall issue a new certificate for the requested server from the trusted CA which contains a name of the same type in the subject name or the SAN extension as appropriate for the reference identifier, and that matches the reference identifier. The evaluator shall configure the requested server to use a valid, supported version and cipher suite combination consistent with the certificate, and provide the new certificate in a valid server certificate message. The evaluator shall initiate a TLS session from the monitored client to the requested server through the TOE and observe that the TSF establishes the TLS session to the requested server.*
- ***Test 3:** (conditional, the TSF supports additional reference identifiers): For each additional reference identifier described in the TSS, the evaluator shall establish a monitored client and requested server to cause the TSF to use the indicated reference identifier and issue a certificate for a server from the trusted CA that contains a name of the same type in the subject name or the SAN extension as appropriate for the reference identifier, but that does not match the reference identifier. The evaluator shall configure the requested server to use a valid, supported version and cipher suite combination consistent with the certificate, and provide the new certificate in a valid server certificate message. The evaluator shall initiate a TLS session between the monitored client and the server through the TOE and observe that the TSF does not establish a valid TLS session to the requested server.*
- ***Test 4:** The evaluator shall perform the following wildcard tests with each type of reference identifier based on DNS name types. This test is not intended for reference identifiers using*

IP addresses. The support for wildcards is intended to be optional. If wildcards are supported, the first, second, and third tests below shall be executed. If wildcards are not supported, then the fourth test below shall be executed. For each test, the evaluator shall establish a monitored client and requested server, issue the requested server a valid certificate with the specified identifier from the trusted CA, and configure the server to use a valid version and cipher suite combination consistent with the certificate. The evaluator shall configure the monitored client and requested server in such a way that causes the TSF to establish the indicated reference identifiers. For each certificate identifier presented and for each reference identifier specified, the evaluator shall initiate a TLS session between the monitored client and requested server through the TSF, causing the TSF to attempt to match the presented identifier to the established reference identifier and observe the indicated result:

- *Test 4.1: (conditional, the TSF supports wildcards): The evaluator shall use a server certificate containing a wildcard that is not in the left-most label of the presented identifier (e.g., foo.*.example.com) and verify that the connection fails.*
- *Test 4.2: (conditional, the TSF supports wildcards): The evaluator shall use a server certificate containing a wildcard in the left-most label but not preceding the public suffix (e.g., *.example.com). The evaluator shall cause the reference identifier to have a single left-most label (e.g., foo.example.com) and verify that the connection succeeds. The evaluator shall cause the reference identifier to be without a left-most label as in the certificate (e.g., example.com) and verify that the connection fails. The evaluator shall cause the reference identifier to have two left-most labels (e.g., bar.foo.example.com) and verify that the connection fails.*
- *Test 4.3: (conditional, the TSF supports wildcards): The evaluator shall use a server certificate containing a wildcard in the left-most label immediately preceding the public suffix (e.g., *.com). The evaluator shall cause the reference identifier to have a single left-most label (e.g., foo.com) and verify that the connection fails. The evaluator shall cause the reference identifier to have two left-most labels (e.g., bar.foo.com) and verify that the connection fails.*
- *Test 4.4: (conditional, the TSF does not support wildcards): The evaluator shall use a server certificate containing a wildcard in the left-most label (e.g., *.example.com). The evaluator shall cause the reference identifier to have a single left-most label (e.g., foo.example.com) and verify that the connection fails.*

FCS_TTTC_EXT.1.3

TSS

The evaluator shall ensure that the TSS describes the TSF's behavior for certificate validation results, including any dependencies on the configured TLS session establishment policy for establishing a TLS session when revocation information is not available, as indicated in the selection for FIA_X509_EXT.2.2.

Guidance

If the TSS indicates that the TLS session establishment policy is used to determine the TSF's behaviour for establishing a TLS session for through-traffic processing when certificate revocation information is not available, the evaluator shall validate that the AGD guidance includes instructions to configure the allowances to allow or not allow such connections.

Tests

Using the setup for test 1 of FCS_TTTC_EXT.1.1, the evaluator shall establish one or more trusted subordinate CAs by issuing them valid CA certificates from the trusted CA. The evaluator shall establish a certificate status capability for both the trusted subordinate CAsand the trusted CA that uses a method supported by the TSF. The evaluator shall also establish an untrusted CA to use a self-signed CA certificate not loaded into the TSF trust store. The evaluator shall establish one or more requested servers to use a valid TLS version and cipher suite combination and to respond using valid TLS handshake messages except for the certificate message and certificate verify messages as described in each test. The evaluator shall issue certificates for the following tests to the requested server that have the indicated failures, initiate a TLS session from a monitored client through the TOE to the requested server presenting the certificate with the indicated failures, and verify that the TSF terminates the TLS handshake with the requested server:

- ***Test 1:** The evaluator shall issue a valid certificate for the requested server from the untrusted CA. The evaluator shall confirm that the TSF rejects the TLS session with the requested server when it presents a valid certificate message and certificate verify message using the certificate issued by the untrusted CA.*
- ***Test 2:** The evaluator shall issue a valid certificate for the requested server by the subordinate CA, but not load it into the TSF trust store, and shall ensure the requested server does not provide the subordinate CA in the certificate chain. The evaluator shall confirm that the TSF rejects the TLS session with the requested server when the server presents a valid certificate message and certificate verify message using the certificate that does not properly chain to a trusted root.*
- ***Test 3:** The evaluator shall establish a valid certificate for the requested server issued by the subordinate CA, and establish valid revocation information from the trusted subordinate CA using a supported mechanism for end-entity certificates, indicating the requested server's certificate is revoked. The evaluator shall ensure the subordinate CA is included in the certificate chain provided by the requested server and the revocation information is*

available. The evaluator shall confirm that authentication fails.

- ***Test 4:*** *The evaluator shall issue a valid certificate for the requested server from the subordinate CA, and establish valid revocation information from the subordinate CA using a supported mechanism for endentity certificates, indicating the requested server's certificate is revoked. The evaluator shall ensure the subordinate CA is included in the certificate chain provided by the requested server and ensure the revocation information is not available to the TSF. The evaluator shall confirm that the default behavior for revocation information not available is performed by the TSF. If this behavior is configurable (the first item is claimed in the first selection for FIA_X509_EXT.2.2), the evaluator shall in turn follow AGD documentation to configure the TSF for each response, and initiate the TLS session from the monitored client to demonstrate the TSF performs the configured behavior.*
- ***Test 5:*** *The evaluator shall issue a valid certificate for the requested server from the subordinate CA, and generate valid revocation information from the trusted subordinate CA using a supported mechanism for end-entity certificates, indicating the requested server's certificate is valid, and generate valid revocation information from the trusted CA using a supported mechanism for CA certificates, indicating the subordinate CA's certificate is revoked. The evaluator shall ensure the subordinate CA is included in the certificate chain provided by the requested server and all revocation information is available. The evaluator shall confirm that authentication fails.*
- ***Test 6:*** *The evaluator shall issue a valid certificate for the requested server from trusted subordinate CA, and generate valid revocation information from the trusted subordinate CA using a supported mechanism for end-entity certificates indicating the requested server's certificate is valid, and generate valid revocation information from the trusted CA using a supported mechanism for CA certificates, indicating the subordinate CA's certificate is revoked. The evaluator shall ensure the subordinate CA is included in the certificate chain provided by the requested server and the revocation information from the subordinate CA is available, but revocation information from the trusted CA is not available to the TSF. The evaluator shall confirm that the default behavior for revocation information not available is performed by the TSF. If this behavior is configurable (the first item is claimed in the selection for FIA_X509_EXT.2.2), the evaluator shall, in turn, follow AGD documentation to configure the TSF for each response, and initiate the TLS session from the monitored client to demonstrate the TSF performs the configured behavior.*
- ***Test 7:*** *The evaluator shall issue a valid certificate from the trusted CA for the requested server that expires prior to initiating the TLS session from the monitored client, and generate revocation information indicating the requested server's certificate is not revoked. The evaluator shall initiate a TLS session from the monitored client to the requested server through the TOE after the certificate has expired, and ensure the certificate status information from the trusted CA is available to the TSF. The evaluator shall observe that the TSF fails to establish the TLS connection with the requested server, demonstrating that a server using a certificate which has passed its expiration date results in an authentication failure.*
- ***Test 8:*** *The evaluator shall establish a new subordinate CA from the trusted CA, by issuing the subordinate CA a certificate that expires prior to initiating the TLS session from the monitored client. The evaluator shall issue a valid certificate for the requested server from the subordinate CA but which does not expire prior to initiating the TLS session from the monitored client and generate valid revocation information using supported methods indicating both the subordinate CA and the server's certificate are not revoked. The evaluator shall initiate a TLS session from the monitored client to the requested server through the TOE and observe that the TLS session between the TSF and requested server fails, demonstrating that a server using a valid certificate (not yet expired) issued by a subordinate CA that has passed its expiration date results in an authentication failure.*

FCS_TTTC_EXT.1.4

TSS

The evaluator shall ensure that the TSS includes a description of cipher suite dependence on the TLS session establishment policy allowances and that the ordering of cipher suites within a Client Hello sent by the TSF to a requested server is in accordance with FCS_TTTC_EXT.1.4.

Guidance

The evaluator shall ensure that the AGD guidance documents include instructions on configuring the TLS session establishment policy to restrict the inclusion of cipher suites in a Client Hello to a particular requested server for through-traffic processing.

Tests

Setup: The evaluator shall establish one or more monitored clients and requested servers that are configured to pass TLS sessions through the TOE, and configure the TLS session establishment policy to use the inspection operation for these clients and servers, allowing negotiation of TLS 1.2, but not any other version, and allowing only a subset of cipher suites indicated in FCS_TTTC_EXT.1.1 consisting of a single cipher suite supported for each supported TLS version. The evaluator shall issue certificates for the servers that are valid in accordance with FIA_X509_EXT.1/STIP, and install the appropriate trust anchors within the TSF to validate the certificates (the trusted CA). Additional configuration instructions for the monitored client, the requested server or the server's certificate are indicated in each of the tests.

- ***Test 1:*** *For each supported version other than TLS 1.2, the evaluator shall configure a*

server requested by a monitored client to negotiate the version and an allowed cipher suite for that version, regardless of the Client Hello message received. The evaluator shall, in turn, establish a TLS connection from the monitored client to the requested server through the TOE and observe that the TSF sends a Client Hello to the requested server that includes the allowed version and cipher suites, in the order indicated in FCS_TTTC_EXT.1.1. The evaluator shall confirm that the requested server sends the TOE a Server Hello indicating the configured version and cipher suite, and confirm that the TSF responds by terminating the TLS handshake with the requested server.

- ***Test 2:*** *The evaluator shall follow AGD guidance to reconfigure the TLS session establishment policy to allow any supported version to the requested servers, but only allow the subset of cipher suites as indicated in the setup. For each supported version, the evaluator shall configure the requested server to negotiate the version and a valid cipher suite for that version which is included in FCS_TTTC_EXT.1.1, but not allowed for the requested server, asin the setup, regardless of the Client Hello received. The evaluator shall in turn initiate a TLS session from the monitored client to the requested server configured for the supported version, through the TOE. The evaluator shall observe that the Client Hello generated by the TSF specifies version 1.2 and the allowed cipher suites in the order indicated in FCS_TTTC_EXT.1.1. The evaluator shall confirm that the server sends the TOE a Server Hello message as configured and confirm that the TSF responds by terminating the TLS handshake with the requested server.*

FCS_TTTC_EXT.5 Thru-Traffic TLS Inspection Client Support for Supported Groups Extension

FCS_TTTC_EXT.5.1

The TSF shall present the Supported Groups Extension in the Client Hellow with the supported groups **[selection:**

- ***secp256r1*****,**
- ***secp384r1*****,**
- ***secp521r1*****,**
- ***ffdhe2048(256)*****,**
- ***ffdhe3072(257)*****,**
- ***ffdhe4096(258)*****,**
- ***ffdhe6144(259)*****,**
- ***ffdhe8192(260)*****,**
- ***[assignment: other supported curves]]***

] .

Application Note: Since support for all of the cipher suites listed in FCS_TTTC_EXT.1.1 is required, at least one of the curves and one of the finite field groups must be chosen by the ST author as appropriate for the cipher suites and the implementation.

If additional elliptic curves are supported, ST author should describe the elliptic curve parameters for each supported elliptic curve in the assignment in accordance with RFC 7919. No additional Diffie-Hellman groups should be claimed in the assignment.

The Supported Groups Extension was previously referred to as the Supported Elliptic Curves Extension and is described in RFC 7919.

Since a requested server session might not adhere to RFC 7919 processing rules, the TOE should accept additional DH groups that might be presented in the requested server's key exchange message.

Evaluation Activities

FCS_TTTC_EXT.5
TSS
The evaluator shall check the TSS and ensure that it describes the supported groups extension. The evaluator shall ensure the TSS describes any configurable aspects of the use of supported groups, including configuration of allowances controlling the use of curves other than the NIST named curves, secp256r1, secp384r1, or secp521r1, if supported.
Guidance
If the TSS indicates that the TOE must be configured to meet FCS_TTTC_EXT.5.1 requirements for the Supported Elliptic Curves Extension, the evaluator shall verify the AGD guidance includes instructions for configuration of the Supported Elliptic Curves Extension.
Tests

- ***Test 1:*** *The evaluator shall establish a requested server to negotiate a supported version and cipher suite using ECDSA signature and ECDHE key exchange using a custom elliptic curve not included in FCS_TTTC_EXT.5.1, regardless of the Client Hello received. The*

evaluator shall follow AGD guidance to configure the TLS session establishment policy so the TSF inspects traffic to the so configured server from a monitored client. The evaluator shall initiate a TLS session to the requested server from the monitored client through the TOE and observe that the TSF sends a Client Hello to the requested server, and receives the configured server Hello Message from the requested server. The evaluator shall confirm that the TSF terminates the TLS handshake with the requested server.

- ***Test 2:*** *(conditional, the TSF supports additional elliptic curves that are managed via TLS session establishment policy allowances): For each elliptic curve claimed in the assignment of FCS_TTTC_EXT.5.1, the evaluator shall establish a requested server to use the curve in a TLS handshake with a supported version and cipher suite using ECDSA signature and ECDHE key exchange, using the curve. The evaluator shall follow AGD guidance to configure the TLS session establishment policy to perform the inspection operation for TLS traffic to the server and allow the server to negotiate the additional curve. The evaluator shall initiate a TLS request from a monitored client to the server and observe that the Client Hello sent from the TSF to the requested includes the allowed curve. The evaluator shall confirm that the configured server sends a Server Hello message to the TSF that selects the curve, and that the TSF accepts the connection.*
- ***Test 3:*** *(conditional, the TSF supports additional elliptic curves that are managed via TLS session establishment policy allowances): For each elliptic curve claimed in the assignment of FCS_TTTC_EXT.5.1, the evaluator shall establish a requested server to use the curve in a TLS handshake with a supported version and cipher suite using ECDSA signature and ECDHE key exchange, regardless of the Client Hello received. The evaluator shall follow AGD guidance to configure the TLS session establishment policy to perform the inspection operation for TLS traffic to the server, but not allow the server to negotiate the additional curve. The evaluator shall initiate a TLS request from a monitored client to the server and observe that the Supported Groups extension Client Hello sent from the TSF to the requested does not include the curve. The evaluator shall confirm that the configured server sends a Server Hello message to the TSF that selects the curve, and that the TSF terminates the TLS handshake with the requested server.*

FCS_TTTS_EXT.1 Thru-Traffic TLS Inspection Server Protocol

FCS_TTTS_EXT.1.1

The TSF shall implement [*TLS 1.2 (RFC 5246), TLS 1.0 (RFC 2246), and [**selection**: TLS 1.1 (RFC 4346), no other TLS versions]*] as a server to the monitored client that supports the following cipher suites: [

- *TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289*
- *TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289*
- *TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289*
- *TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289*
- *TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288*
- *TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246*
- *TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288*
- *TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246*
- *TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289*
- *TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289*
- *TLS_DHE_RSA_ WITH_AES_256_CCM as defined in RFC 6655*
- *TLS_RSA_ WITH_AES_256_CCM as defined in RFC 6655*
- *TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289*
- *TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289*
- *TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289*
- *TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289*
- *TLS_DHE_RSA_WITH_AES_128_CCM as defined in RFC 6655*
- *TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288*
- *TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246*
- *TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289*
- *TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289*
- *TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289*
- *TLS_RSA_WITH_AES_128_CCM as defined in RFC 6655*
- *TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288*
- *TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246*
- *TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 8422*
- *TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 8422*
- *TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 5246*
- *TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 8422*
- *TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 5246*
- *TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 8422*
- *TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 8422*
- *TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 8422*
- *TLS_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246*
- *TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5426*

- *TLS_RSA_WITH_AES_128_CCM_8 as defined in RFC 6655*
- *TLS_DHE_RSA_WITH_AES_128_CCM_8 as defined in RFC 6655*
- *TLS_DHE_RSA_WITH_AES_256_CCM_8 as defined in RFC 6655*
- *TLS_RSA_WITH_AES_256_CCM_8 as defined in RFC 6655*
- *[**selection**: TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA as defined in RFC 8422, TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA as defined in RFC 5246, TLS_RSA_WITH_3DES_EDE_CBC_SHA as defined in RFC 5246, [**assignment**: other supported cipher suites], no other cipher suites]*

] and also supports functionality for [**selection**:

- *mutual authentication*,
- *session renegotiation*,
- *neither mutual authentication nor session renegotiation*

].

Application Note: TLS version 1.2 and 1.0 must be supported; support for TLS version 1.1 is optional, and should be chosen if the STIP supports it. The list of cipher suites to support is mandatory but includes some selections in order to support legacy clients that may be required by the organization; additional cipher suites can be included in the assignment.

The above list (as instantiated in the ST) limits the cipher suites that may be specified by the TOE when responding to the monitored client. The data encryption and decryption algorithms used in this element are performed in accordance with FCS_COP.1/STIP.

The selection should indicate if mutual authentication and/or session renegotiation is supported. These selections must be the same for both FCS_TTTC_EXT.1.1 and FCS_TTTS_EXT.1.1. If mutual authentication is selected, the requirements in Section B.4 will be included by the ST authors. For this technology, mutual authentication is not desirable on these connections because the STIP will have to issue a certificate representing the client to the requested server, and the server will have to have a trust anchor for that certificate. If session renegotiation is selection, FCS_TTTS_EXT.4 in section B.5 will be included by the ST authors.

The data encryption and decryption algorithms used in this element are performed in accordance with FCS_COP.1/STIP.

FCS_TTTS_EXT.1.2

The TSF shall deny connections from clients requesting [SSL 2.0, SSL 3.0, and [**selection**: *TLS 1.1, no other SSL or TLS versions*] for through-traffic processing.

Application Note: All SSL versions are denied regardless of exception specifications. Any TLS versions not selected in FCS_TTTS_EXT.1.1 should be selected here

FCS_TTTS_EXT.1.3

The TSF shall perform key establishment for TLS with a monitored client using [

- *RSA with key size 2048 bits, [**selection**: 1024 bits, 1536 bits, 3072 bits, 4096 bits, no other sizes]*
- *[**selection**:*
  - *Diffie-Hellman parameters of size 2048 bits, [**selection**: 1024 bits, 1536 bits, 3072 bits, 4096 bits, 8192 bits, no other sizes] ,*
  - *Diffie-Hellman groups ffdhe2048, [**selection**: ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192, no other groups]*

  ]
- *EC Diffie-Hellman parameters using elliptic curves [**selection**: secp256r1, secp384r1, secp521r1, [**assignment**: other curves]] and no other curves*

.

Application Note: The selections in this element should indicate all key establishment sizes and/or groups supported.

Evaluation Activities

FCS_TTTS_EXT.1.1
TSS
The evaluator shall check the description of this protocol in the TSS to ensure that the TLS versions and cipher suites supported for establishing a TLS session with a monitored client include those listed in FCS_TTTS_EXT.1.1 and determine if configuration is needed to restrict

the use of other versions or cipher suites. The evaluator shall ensure the TSS description of TLS includes all TLS server handshake messages and error alerts used, and conditions for which error alerts are used.

Guidance

The evaluator shall check the guidance documentation to ensure it contains instructions as indicated in the TSS on configuring the TOE so that the versions and cipher suites used conform with FCS_TTTS_EXT.1.1.

Tests

The evaluator shall establish one or more monitored clients and servers that are configured to pass TLS sessions through the TOE, and configure the TLS session establishment policy to use the inspection operation for these clients and servers with the required versions and cipher suites. The evaluator shall establish certificates for the servers that are valid in accordance with FIA_X509_EXT.1/STIP and install the appropriate trust anchors within the TSF to validate the certificates. Additional configuration instructions for the monitored client, the requested server or the server's certificate are indicated in each of the tests:

- ***Test 1:** For each version and each valid cipher suite for the version, as indicated in FCS_TTTS_EXT.1.1, the evaluator shall configure the monitored client to include the version and a list consisting of a single element specifying the indicated cipher suite in the Client Hello. The evaluator shall follow AGD guidance to configure the TLS session establishment policy to allow the TSF to negotiate the version and cipher suite for that client. The evaluator shall then initiate a TLS session from the so configured monitored client through the TOE to a requested server and observe that a TLS session between the monitored client and the TSF using the specified version and cipher suite is successful.*
- ***Test 2:** For each supported version indicated in FCS_TTTS_EXT.1.1, the evaluator shall select a valid cipher suite in FCS_TTTS_EXT.1.1 and configure a monitored client to present the version and a cipher suite list containing the single cipher suite. The evaluator shall follow AGD guidance to configure the TLS session establishment policy so that use of the cipher suite is not allowed for the client. The evaluator shall initiate a TLS session from the monitored client through the TOE to a requested server and observe that a TLS session between the monitored client and the TSF is denied.*
- ***Test 3:** For each supported version other than TLS 1.2 indicated in FCS_TTTS_EXT.1.1, the evaluator shall configure a monitored client to include the version and a cipher suite list consisting of a cipher suite valid for TLS 1.2 and another valid for the version in its Client Hello. The evaluator shall follow AGD guidance to configure the TLS session establishment policy to only allow the client to use TLS 1.2. The evaluator shall initiate a TLS session from the monitored client to a requested server through the TOE and observe that a TLS session between the monitored client and the TSF is not established.*
- ***Test 4:** For each supported version indicated in FCS_TTTS_EXT.1.1, the evaluator shall configure a monitored client to include the version and a cipher suite list consisting of a single TLS_NULL_WITH_NULL_NULL cipher suite. The evaluator shall follow AGD guidance to configure the TLS session establishment policy to allow any version and cipher suite for the client. The evaluator shall initiate a TLS session from the monitored client to a requested server through the TOE and observe that the TLS session between the monitored client and the TSF is denied.*

FCS_TTTS_EXT.1.2

TSS

The evaluator shall verify that the TSS contains a description of the denial of SSL versions and TLS versions consistent with the selections in FCS_TTTS_EXT.1.2 and determine if configuration is needed to restrict the use of those versions.

Guidance

The evaluator shall check the guidance documentation to ensure it contains instructions on configuring the TOE as indicated in the TSS so that the versions indicated in FCS_TTTS_EXT.1.2 are denied.

Tests

For each SSL or TLS version indicated in FCS_TTTS_EXT.1.2, the evaluator shall configure a monitored client to include the version in its Client Hello. The evaluator shall initiate a TLS session from the monitored client to a requested server through the TOE and observe that a TLS session between the monitored client and the TOE is not established.

FCS_TTTS_EXT.1.3

TSS

The evaluator shall verify that the TSS describes the TOE's supported key agreement parameters for a server key exchange message with a monitored client to ensure TOE supports the required key agreement parameters and can be limited to use only those indicated in FCS_TTTS_EXT.1.3.

Guidance

The evaluator shall check the guidance documentation to ensure it contains instructions on configuring the TOE so that the key exchange parameters used conforms with FCS_TTTS_EXT.1.3.

Tests

The evaluator shall establish one or more monitored clients and servers that are configured to pass TLS sessions through the TOE, and configure the TLS session establishment policy to use the inspection operation for these clients and servers with the required versions and cipher

suites. The evaluator shall establish certificates for valid servers in accordance with FIA_X509_EXT.1/STIP and install the appropriate trust anchors within the TSF to validate the certificates. Additional configuration instructions for the monitored client, the requested server, or the server's certificate are indicated in each of the tests.

- ***Test 1:*** *For each of the key parameter selections in FCS_TTTS_EXT.1.3, the evaluator shall configure a monitored client to use a valid supported version and cipher suite combination that supports the key parameter, and follow AGD guidance to configure the TSF to use a cipher suite supporting the parameters. The evaluator shall initiate a TLS session from the monitored client to a requested server through the TOE, and observe that a TLS session between the monitored client and the TLS uses the key parameters and is successful.*
- ***Test 2:*** *The intent of this test is to show that the TSF properly handles unexpected KeyExchange messages from a client that does not agree with the negotiated cipher suite.*

  For each of the key parameter selections claimed for FCS_TTTS_EXT.1.3, the evaluator shall configure a monitored client and follow AGD guidance to configure the TSF to use a cipher suite supporting the parameters.

  For each such configuration, the evaluator shall, in turn, initiate a number of TLS sessions from the monitored client to a requested server through the TOE, interrupting the TLS exchanges after receiving a server certificate from the TSF and sending the specified client KeyExchange message and observe the results as indicated below:

  a. *For a cipher suite that uses RSA for key transport, the evaluator shall, in turn, perform each of the following:*
     i. *In the first instance of test 2.a, the evaluator shall send a KeyExchange message of RSA type with the EncryptedPreMasterSecret field consisting of a randomly generated value of size equal to the size of the EncryptedPreMasterSecret expected in the key parameter. The evaluator shall observe that the TSF sends a fatal TLS alert message and note the specific alert type received agrees with the TSS description of error messages.*
     ii. *In the second instance of test 2.a, the evaluator shall send a KeyExchange message of RSA type with the EncryptedPreMasterSecret field consisting of a randomly generated value of size 1024 bits. The evaluator shall observe that the TSF sends a fatal TLS alert message.*
     iii. *In the third instance of test 2.a, the evaluator shall send the TSF a KeyExchange message of DHE type containing a randomly generated ClientDiffieHellmanPublic value of size 2048 bits, and observe that the TSF sends a fatal TLS alert message and notes whether the error message is different than that received test 2.a.i.*
     iv. *in the fourth instance of test 2.a, the evaluator shall send the TSF a KeyExchange message of ECDH type, containing a random point on a curve supported by the TSF, in a EC point format supported by the TSF. The evaluator shall observe that the TSF sends a fatal TLS alert message and notes whether the error message is different than that received in test 2.a.i.*

  If the alert messages in 2.a.iii and 2.a.iv are identical to that received in 2.a.i, the evaluator shall attempt to verify that the errors are a result of the unexpected KeyExchange message, and not just due to an invalid finished message. It might be necessary to configure additional (debug) logs to be generated by the TSF or examine detailed behavior of the TSF to distinguish unexpected KeyExchange messages from other errors.

  b. *For a cipher suite that uses ephemeral DH key establishment:*
     i. *In the first instance of test 2.b, the evaluator shall modify a byte in the ClientDiffieHellmanPublic value produced by the client, send the modified KeyExchange message to the TSF, and observe that the TSF sends a fatal TLS alert message and note that the specific error message agrees with the TSS description of error messages.*
     ii. *In the second instance of test 2.b, the evaluator shall ensure the TSF is not configured to request client authentication. The evaluator shall send a KeyExchange message consisting of a null value, specifying an implicit Client DiffieHellman Public key. The evaluator shall send the modified KeyExchange message to the TSF, and observe that the TSF sends a fatal TLS alert message and notes whether the error message is different than that received in test 2.b.i.*
     iii. *In the third instance of test 2.b, the evaluator shall send the TSF a KeyExchange message of RSA type, containing a randomly generated EncryptedPreMasterSecret value of size 2048 bits. The evaluator shall observe that the TSF sends a fatal TLS alert message and notes whether the error message is different than that received in test 2.b.i.*
     iv. *(conditional, the TSF supports client authentication): In the fourth instance of test 2.b, the evaluator shall configure the TSF to request client authentication. After the TSF sends the client certificate request message, the evaluator shall send the TOE a valid client certificate message followed by a KeyExchange message of ECDH type that contains a random point on a curve supported by the TSF in an ECpoint format supported by the TSF. The evaluator shall observe that the TSF sends a fatal TLS alert message and notes whether the error message is different than that received in test 2.b.i.*

  If the error messages in 2.b.ii, 2.b.iii or 2.b.iv are identical to that provided in 2.b.i, the

evaluator shall attempt to verify that the errors are a result of the unexpected KeyExchange message, and not just due to an invalid finished message. It might be necessary to configure additional (debug) logs to be generated by the TSF or examine detailed behavior of the TSF to distinguish unexpected KeyExchange messages from other errors.

c. *If the cipher suite uses ephemeral ECDH key establishment:*
   i. *. In the first instance of test 2.c, the evaluator shall replace the EC point in the KeyExchange message produced by the client with a random point on the curve specified by the TSF's Server key exchange message, using the same EC point format used in the client's expected KeyExchangeMessage. The evaluator shall observe that the TSF sends a fatal TLS alert message and the specific alert message agrees with the error message description in the TSS.*
   ii. *In the second instance of test 2.c, the evaluator shall ensure the TSF is not configured to request client authentication, and send the TSF a KeyExchange message consisting of the null value, indicating an implicit client Elliptic Curve Diffie Hellman Public key. The evaluator shall observe that the TSF sends a fatal TLS alert message and notes whether the error message is different than that received in test 2.c.i.*
   iii. *In the third instance of test 2.c, the evaluator shall send the TSF a KeyExchange message of RSA type with a randomly generated 2048-bit value used for the EncryptedPreMasterSecret value. The evaluator shall observer that the TSF sends a fatal TLS alert message and notes whether the error message is different than that received in test 2.c.i.*
   iv. *In the fourth instance of test 2.c, the evaluator shall send the TSF a KeyExchange message of type DH with a randomly generated 2048-bit value for the ClientDiffieHellman value in place of the ephemeral public key. The evaluator shall observe that the TSF sends a fatal TLS alert message and notes whether the error message is different than that received in test 2.c.i.*

   If the error messages received in 2.c.ii, 2.c.iii, or 2.c.iv are identical to that received in 2.c.i, the evaluator shall attempt to verify that the errors are a result of the unexpected KeyExchange message, and not just due to an invalid finished message. It might be necessary to configure additional (debug) logs to be generated by the TSF or examine detailed behavior of the TSF to distinguish unexpected KeyExchange messages from other errors.

d. *(conditional, the TSF supports client authentication): The evaluator shall configure the TSF to request client authentication. For a cipher suite that uses static DH for key transport, the evaluator shall send the TSF a valid client certificate message, followed by a KeyExchange message of DHE type containing a randomly generated ClientDiffieHellmanPublic value of size 2048 bits, and observe that the TSF sends a fatal TLS alert message.*

e. *For a cipher suite that uses static ECDH for key transport, the evaluator shall send the TSF a valid certificate message, followed by a KeyExchange message of ECDHE type, containing a random point on a curve supported by the TSF, in a ECpoint format supported by the TSF, and observe that the TSF sends a fatal TLS alert message.*

- ***Test 3:*** *The intent of this test is to ensure the TSF, when negotiating cipher suites using RSA key transport, responds to invalid RSA KeyExchange messages consistently in order to resist a well-known class of chosen ciphertext attacks against RSA key transport mechanisms, which are especially problematic in TLS 1.0.*

  Initial setup: The evaluator shall establish a monitored client with full debugging and control of the TLS functions to send a TLS Client Hello indicating support for TLS 1.0 and a single cipher suite using RSA key transport. The evaluator shall establish a requested server configured to negotiate TLS 1.0 with the cipher suite indicated by the monitored client. The evaluator shall configure the TSF to inspect traffic between the monitored client and requested server and to allow the version and cipher suite for the client and server, and note this initial configuration for subsequent sub-tests:

  Test 3, part a: *The evaluator shall send a Client Hello from the monitored client to the requested server and observe that the Server Hello from the TSF selects TLS 1.0 and the desired cipher suite in its Server Hello message. The evaluator shall note the size and formatting of pre-master secret input to the client's KeyExchange message, continue the handshake from the client, and confirm that the TSF successfully establishes a TLS connection with the client.*

  Test 3, part b: *The evaluator shall terminate the TLS sessions and restore the TSF, monitored client and requested server to the initial configuration for Test 3 above. The evaluator shall compute the following KeyExchange based on encrypting the following tailored messages with the server's public key, using a random value, ran, of size equal to that of the correctly computed pre-master secret, but having a different value, and properly formatted padding, pad(), of length determined so that the message is of the proper size.*
  - *M1= 0x0002|| pad()||0x00||TLSversion||ran*
  - *M2= 0x4117|| pad()*
  - *M3= 0x0002|| pad()||0x0011*
  - *M4= 0x0002|| pad()*
  - *M5= 0x0002|| pad()||0x00||0x0202||ran*

For each message in turn, the evaluator shall forward the KeyExchange message including the encrypted message to the TSF as part of a complete TLS handshake with the server, and observe the TLS error alert response provided by the TSF. Between each iteration, the evaluator shall terminate any residual TLS sessions, reset any cache, and restore the configuration of the monitored client, requested server, and TOE to its initial configuration for Test 3.

Test 3, part c: *The evaluator shall observe that each error alert response provided by the TSF for the iterations in part b match the description in the TSS and is identical for each message M1 through M5.*

5.2.4 User Data Protection (FDP)

FDP_CER_EXT.1 Certificate Profiles for Server Certificates

FDP_CER_EXT.1.1

The TSF shall implement a certificate profile function for TLS server certificates issued by a CA embedded within the TOE, and shall ensure that issued certificates are consistent with configured profiles.

FDP_CER_EXT.1.2

The TSF shall generate certificates using profiles that comply with requirements for certificates as specified in IETF RFC 5280, “Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile” as refined below. At a minimum, the TSF shall ensure that:

a. The version field shall contain the integer 2.
b. The issuerUniqueID or subjectUniqueID fields are not populated.
c. The serialNumber shall be unique with respect to the issuing Certification Authority.
d. The validity field shall specify a notBefore value that does not precede the current time and a notAfter value that does not precede the value specified in notBefore.
e. The issuer field is not empty and is populated with the **[selection: *Security Administrator, CA Operations Staff*]**-configured CA name.
f. The signature field and the algorithm in the subjectPublicKeyInfo field shall contain the OID for a signature algorithm specified in FCS_COP.1/SigGen in the NDcPP.
g. The following extensions are supported:
    a. authorityKeyIdentifier
    b. keyUsage
    c. extendedKeyUsage
    d. certificatePolicy
    e. [**selection**: *subjectKeyIdentifier*, *basicConstraints*, *cRLDistributionPoints*, *authorityInfoAccess*, *no other extensions*]
h. A subject field containing a null Name (e.g., a sequence of zero relative distinguished names) is accompanied by a populated critical subjectAltName extension.
i. The authorityKeyIdentifier extension in any certificate issued by the TOE must be populated and must be the same as the subjectKeyIdentifier extension contained in the TOE’s embedded CA’s signing certificate
j. Populated keyUsage and extendedKeyUsage fields in the same certificate shall contain consistent values reflecting exclusive TLS server use as follows:

keyUsage	extendedKeyUsage
digitalSignature	serverAuth
digitalSignature, keyEncipherment	serverAuth
digitalSignature,keyAgreement	serverAuth

k. [**selection**: *The subjectKeyIdentifier extension is populated with a value unique for each public key contained in a certificate issued by the TSF, no other constraints*]

Application Note: RFC updates to RFC 5280 are included in this requirement. The inclusion of the cRLDistributionPoints and authorityInfoAccess extensions depend on the selections made in FDP_CSI_EXT.1.3.

Uniqueness for the subject key identifier (item k above) is specific to the instance of the embedded CA. The same configured CA should not issue certificates with different public keys having the same subject key identifier.

If subjectKeyIdentifier is chosen in the selection in item g, then the ST author selects the first selection in item k; otherwise, select “no other constraints.”

FDP_CER_EXT.1.3

The TSF shall implement the following rules for populating certificate fields based on constraints imposed by the TOE’s embedded CA’s signing certificate:

- The validity field shall specify a notAfter time that does not exceed the notAfter time of the CA’s signing certificate.
- The issuer field identifies the [**selection**:
  - *subject,*
  - *[**assignment**: **[selection: Security Administrator, CA Operations Staff]**-assigned identifying information]*

  ] of the CA's signing certificate.
- [**selection**:
  - *The subject name is limited by name constraints specified in the CA’s signing certificate,*
  - *[**assignment**: list of rules],*
  - *no other rules*

  ]

FDP_CER_EXT.1.4

The TSF shall implement the following rules for populating certificate fields based on the validated certificate and constraints imposed by the **[selection: *Security Administrator, CA Operations Staff***]:

a. The Subject/Subject Alternative Name shall be copied from validated server certificate.
b. The notBefore field shall not precede the notBefore field of the validated server certificate.
c. The notAfter field shall not exceed the notAfter field of the validated server certificate.
d. The notAfter field shall not exceed the current time by more than a maximum validity duration value as configured by a **[selection: *Security Administrator, CA Operations Staff***] user.
e. If the basicConstraints field is configured to be present, it shall be populated with the value cA=False
f. The subject public key shall be generated in accordance with FCS_CKM.1.1 **in the NDcPP.**
g. [**selection**:
   - *policy OID/policy mapping fields are populated in accordance with [**assignment**: a **[selection: Security Administrator, CA Operations Staff]** configured mapping from validated server certificate values to one or more stated policy OIDs],*
   - *[**assignment**: list of rules],*
   - *no additional rules*

   ]

Application Note: It is preferred that a new public key be generated each time a certificate is generated.

Evaluation Activities

FDP_CER_EXT.1
TSS
The evaluator shall examine the TSS to ensure it describes the certificate profile function in accordance with FDP_CER_EXT.1.1 The TSS shall describe how certificate profiles are configured and then selected to issue certificates in accordance with FDP_CER_EXT.1.2.
Guidance
The evaluator shall examine the operational guidance to ensure that instructions are available to configure certificate profiles used for certificate generation in accordance with this requirement.
Tests
The evaluator shall perform the following tests:

- ***Test 1:** The evaluator shall configure a certificate profile using the available guidance, and establish a server with a certificate that satisfies FDP_CER_EXT.1.2 items a, b, e, f, h, i, j, and k, has valid values in all extensions in item g (a-e), and passes all certificate validation criteria as a TLS server certificate (having extended key usage field of server authentication) in FIA_X509_EXT.1/Rev. The evaluator shall establish a monitored client and request a TLS session to the server through the TOE so that the inspection operation is implemented, and then examine the certificate received at the client from the TOE to ensure it matches the configured certificate profile.*

- ***Test 2:*** *The evaluator shall specifically examine the certificate generated in Test 1 and compare it to both the embedded CA's certificate and the requested server's certificate to ensure that it satisfies all field constraints in FDP_CER_EXT.1.2, FDP_CER_EXT.1.3, and FDP_CER_EXT.1.4 as configured in the certificate profile.*
- ***Test 3:*** *The evaluator shall conduct the following tests by establishing a server with certificate identical to that used in Test 1, except for the differences described as follows (each in turn). The evaluator shall make any configuration changes to the TOE as indicated, establish a monitored client, and submit a TLS request for the server through the TOE so that the inspection operation is performed, and observe the certificate received at the monitored client has the indicated features:*
  - ***notBefore field test:*** *The evaluator shall assign a notBefore value in the established server certificate that precedes both the current time and the value of notBefore field in the TOE's embedded CA's certificate, and observe that the generated certificate has a notBefore value that does not precede the current time.*
  - ***notAfter field test a:*** *The evaluator shall configure the maximum validity duration so that the notAfter value of the TOE's embedded CA certificate does not exceed the current time by more than the maximum validity duration. The evaluator shall assign a notAfter value in the established server certificate that exceeds the current time by more than the maximum validity period, and observe that the notAfter field of the generated certificate has a notAfter value that does not exceed the notAfter value of the embedded CA's certificate.*
  - ***notAfter field test b:*** *The evaluator shall configure the maximum validity duration so that the notAfter value in the TOE's embedded CA certificate exceeds the current time by more than maximum validity duration, assign a notAfter value in the established server certificate that exceeds the notAfter value in the TOE's embedded CA's certificate, and observe that the notAfter value of the generated certificate does not exceed the current time by more than the maximum validity duration.*
  - ***notAfter field test c:*** *The evaluator shall assign a notAfter value in the established server certificate that precedes both the notAfter value in the TOE's embedded CA's certificate, and the current time plus the maximum validity duration, and observe that the generated certificate has a notAfter value that does not exceed the notAfter value of the established server's certificate.*
  - ***keyUsage field test:*** *The evaluator shall assign a KeyUsage value in the established server certificate that indicates additional usage indicators (e.g., keyCertSign) and observe that generated certificate has only the digitalSignature and/or keyEncipherment indicators.*
  - ***extendedKeyUsage field test a:*** *The evaluator shall omit the extendedKeyUsage field in the established server certificate and observe that the generated certificate contains the extendedKeyUsage field with value indicating only TLS server authentication.*
  - ***extendedKeyUsage field test b:*** *The evaluator shall populate the extendedKeyUsage field in the established server's certificate to indicate both TLS server authentication and code signing, and observe that the generated certificate only indicates TLS server authentication.*
  - ***extendedKeyUsage field test c:*** *The evaluator shall populate the extendedKeyUsage field in the established server's certificate to indicate any usage, and observe that the generated certificate only indicates TLS server authentication.*

FDP_CER_EXT.2 Certificate Request Matching of Server Certificates

FDP_CER_EXT.2.1

The TSF shall establish and record a linkage from validated certificates to issued certificates.

Application Note: This requirement ensures that the TOE provides linkage between TLS server certificates validated during a TLS session establishment by the TOE and resulting certificates issued by the TOE to represent the requested server (or monitored client if supported). In terms of Certification authority operations, an automatically approved certificate request is implied by the validated certificate and the configured TLS session establishment policy identified in FDP_TEP_EXT.1.

Evaluation Activities

FDP_CER_EXT.2
TSS
The evaluator shall examine the TSS to ensure it describes the linkage between submitted requests and issued certificates and indicates where this linkage is recorded.
Guidance
The evaluator shall examine the operational guidance to ensure it contains instructions for how to trace a submitted request to an issued certificate and vice versa via the TOE's interface.

Tests
The evaluator shall configure a certificate profile using the available guidance and establish a server with a server certificate which is consistent (would allow the CA to issue a certificate) with the profile. The evaluator shall establish a client and request a TLS session with the server so that the inspection operation is selected. The evaluator shall follow the administrative guidance for determining the linkage and verify that it provides linkage between the validated server certificate and issued certificate.

FDP_CER_EXT.3 Certificate Issuance Rules for Server Certificates

FDP_CER_EXT.3.1

The TSF shall issue certificates in response to a validated server certificate according to the following rules: The issued certificate is in compliance with a current certificate profile defined in accordance with FDP_CER_EXT.1 and

- The TLS session establishment policy is configured to allow inspection of TLS sessions between monitored clients and a requested server authenticated to the TSF by the validated certificate,

[**selection**:

- *A valid certificate for the same subject is not present in cache,*
- *The embedded CA certificate's name space allows issuance of a certificate that represents the authenticated server*

].

FDP_CER_EXT.3.2

The TSF shall reject all certificate requests originating external to the TOE.

Evaluation Activities

FDP_CER_EXT.3
TSS
The evaluator shall examine the TSS to ensure it describes the certificate issuance rules, and verify that any interfaces available for external certificate requests (CMC, EST, PKCS#10 or any other request format) are identified.
Guidance
The evaluator shall examine the operational guidance to ensure that it contains instructions for any configuration aspects of any certificate issuance approval function and the steps needed to prevent receipt and approval of external requests.
Tests
The evaluator shall perform the following tests:

- ***Test 1:*** *(conditional, the TSF has one or more interfaces that could be used to receive external certificate requests): For each interface that can be used to receive external certificate requests, the evaluator shall configure the certificate issuance approval function in accordance with the operational guidance. The evaluator shall create a certificate request and submit it to the TOE. The evaluator shall access the TOE using the defined interface and verify that the submitted request is rejected.*
- ***Test 2:*** *The intent of this test is to exercise a representative set of SSL/TLS inspection proxy rules for the supported features of the TLS session establishment policy and demonstrate certificates are generated by the TSF only when the inspection operation is authorized.*

  The evaluator shall follow AGD guidance to configure a set of rules for the TLS session establishment policy that exercises the inspection operation, bypass operation, and block operation for a representative sample of supported monitored client requested server abstractions as indicated in FDP_TEP_EXT.1.4.

  The evaluator shall further configure rules that specify allowances restricting a subset of the monitored client, requested server abstractions associated with the inspection operation to specific TLS versions, cipher suites, supported groups, and other constraints as indicated in the selection of FDP_TEP_EXT.1.5.

  The evaluator shall establish TLS servers with certificates issued by an external certification authority, such that for each rule specified, at least one server has attributes satisfying the rule. The evaluator shall establish monitored clients so that for each rule, at least one monitored client has attributes satisfying the rule. If client authentication is supported, the evaluator shall issue certificates to monitored clients from a certification authority trusted by the TSF as required to exercise the rules.

  For each rule restricting the TLS allowances, the evaluator shall establish monitored clients, requested servers, and certificates as necessary that match rules associated with the inspection operation, but violate the allowances for the requested server and monitored client pair.

  The evaluator shall initiate TLS requests from monitored clients through the TOE, to requested servers to exercise the rules. The evaluator shall observe the resulting logs to

confirm the rule is exercised as intended.

For each instance where the rule is associated with the inspection operation and no TLS allowances are violated, the evaluator shall inspect the TLS server certificate message sent from the TOE to the monitored client and confirm the TOE's embedded CA issues the certificate. The evaluator shall search the certificate repository to identify the issued certificate associated with the requested server and that the certificate in the repository matches the certificate sent to the monitored client.

For each instance where the rule is associated with the inspection operation but the TLS allowances are violated, the evaluator shall inspect the TSF logs to confirm the session was blocked. When the server TLS allowances associated with the Client Hello received from the monitored client (version, cipher suites), with the Server Hello received from the requested server (version, cipher suite, supported groups and critical extensions), or with the requested server certificate validation (including certificate revocation information not available when inspection is not allowed), are violated, the evaluator shall search the certificate repository to ensure no certificate matching the subject field in the requested server's certificate is associated to the current session, and search the certificate repository to ensure no certificate matching any of the names in the subject alternate name extension in the requested server's certificate is associated with the current session.

Note: Certain allowances (associated with key exchange messages or client certificate messages received after the server certificate message is sent) may only be determined to be violated after the TSF issues a certificate for the requested server.

For each instance where the rule is associated with the bypass operation, the evaluator shall inspect the TLS server certificate message sent from the TOE to the monitored client, and the TLS server certificate message sent from the requested server to the TOE. The evaluator shall: verify that the certificate sent to the monitored client matches the certificate sent from the requested server exactly, confirm that the certificate issuer indicated in the certificate is the CA trusted by the TOE, and not the TOE's embedded CA, and search the certificate repository for the certificate to confirm the certificate is not present as an issued certificate.

For each instance where the rule is associated with the block operation, the evaluator shall search the certificate repository for any certificate matching the subject field of the requested server's certificate and observe that no certificate was issued in response to the request. The evaluator shall also search the certificate repository for any certificate matching any of the names included in the requested server's subject alternate name extension and observe that no certificate was issued in response to the request.

- ***Test 3:*** *(conditional, the TSF supports caching of issued certificates): The evaluator shall configure the TSF to retain certificates in the cache, and initiate a TLS session from a monitored client to a requested server as in Test 2 where the monitored client requested server combination matches a rule associated with the inspection operation without allowance violations. The evaluator shall confirm that the certificate issued by the TOE's embedded CA is contained in the certificate repository. The evaluator shall then establish a second monitored client for which the second monitored client and same requested server also match a rule associated with the inspection operation without allowance violations. The evaluator shall initiate a TLS session from the second client to the same requested server, observe logs to verify that the inspection operation was performed, and search the certificate repository to confirm that a new certificate for the request was not issued.*

FDP_CSIR_EXT.1 Certificate Status Information Required

FDP_CSIR_EXT.1.1

The TSF shall [**selection**: *generate certificate status information*, *only issue certificates with validity period of less than [24 hours]*].

Application Note: Based on the selection, the ST author must choose the appropriate requirements from Appendix B.1 of this PP-Module.

The ST should specify whether certificate status information is generated. If the TSF can be configured so the validity of issued certificates is longer than 24 hours, certificate status information must be able to be generated.

Certificate policies associated with the issuance of TLS server certificates imply that certificates issued by the TSF must be revoked within a certain time period of discovering they do not properly represent the asserted subject. Certificate status information is not required if the validity period of any issued certificate is less than the time in which this status information must be provided. Even for emergency revocations, this time period is typically greater than 24 hours.

Evaluation Activities

FDP_CSIR_EXT.1

TSS
The evaluator shall examine the TSS to ensure it describes whether certificate status information is provided.
Guidance
The evaluator shall examine the operational guidance to ensure that it contains instructions for any configuration aspects of the validity period that are necessary for the TSF to operate in compliance with this requirement.
Tests
If the TSF provides certificate status information, testing for this functionality is performed under the certificate status information requirements that are claimed. If the TSF does not provide certificate status information and instead issues certificates with a lifetime under 24 hours, the evaluator shall perform the following test:

The evaluator shall follow AGD documentation to configure the TSF in compliance with this SFR. The evaluator shall establish a monitored client and a requested server whose certificate is valid for one year, and configure the TSF to inspect TLS traffic between the monitored client and requested server. The evaluator shall initiate a TLS session between the monitored client and requested server. The evaluator shall observe that a certificate is received at the monitored client, which is issued by the TSF and shall verify that its validity is less than 24 hours.

FDP_PPP_EXT.1 Plaintext Processing Policy

FDP_PPP_EXT.1.1

The TSF shall enforce the TLS plaintext processing policy on information flows containing plaintext produced by inspection processing of the TOE between TLS session termination points and [**selection**: *distinct internal inspection processing functional components*, *internal inspection processing functional components and an interface to external inspection processing environment*]

Application Note: This element identifies the policy (TLS plaintext processing) that is applied to decrypted TLS session data received by the TSF via an external interface for which the TLS session establishment policy, FDP_TEP_EXT.1, determines inspection processing is authorized, resulting in the exposure of the underlying plaintext associated to the TLS session. Information flows containing such data are referred to as TLS session threads.

Every network packet decrypted under the TLS session establishment policy inspection operation is associated to a TLS session thread and has the ruleset that expresses this policy applied between each distinct inspection processing functional component, including the points where TLS encryption/decryption occurs. This PP-Module allows both internal and external inspection processing functional components. Internal inspection processing components, if supported, range from simple routing functions that determine whether to abort inspection processing of a TLS session thread based on an identifier, to complicated intrusion detection/prevention functions. External inspection processing components, if supported, are accessed via a controlled interface of the TOE to a protected computing environment, considered as part of the operational environment.

FDP_PPP_EXT.1.2

The TSF shall allow the definition of TLS plaintext processing policy rules using [**assignment**: *entity attributes of the requested server*], [**assignment**: *indicators of inspection processing results*] and distinct interfaces.

Application Note: This element identifies the various attributes that are applicable when constructing rules to be enforced by this requirement. The attributes to be included in this requirement include those which are exposed only for TLS sessions undergoing inspection processing in accordance with FDP_TEP_EXT.1 after the TLS application payload is decrypted, or can simply be the thread indicator to implicitly include attributes obtained by the TLS session establishment policy. Indicators can include specific error alerts from an internal inspection processing functional component, or can be a timeout resulting from an inspection processing functional component blocking traffic requiring no explicit signaling.

FDP_PPP_EXT.1.3

The TSF shall allow the following operations to be associated with the plaintext processing policy: permit, block, and [**selection**: *bypass*, *no other operation*], with the capability to log the operation.

Application Note: This element defines the operations that can be associated with rules used to manage inspection processing of TLS session threads. Permit allows the information flow to continue between inspection processing functional components; bypass indicates that the information flow is not processed by the processing component, but is forwarded to either the TLS encryption/decryption

buffer, or the next plaintext inspection functional processing component; block drops subsequent information flows associated to the TLS session thread and informs the TLS session establishment policy to transition the TLS session to a Block Operation for subsequent TLS messages to or from the monitored client or requested server. It is permissible to use timeouts as indicators between inspection processing functional components or between the TLS plaintext processing policy and the TLS session establishment policy. Note this requirement does not specify the behavior of the inspection processing functional components, as this functionality is out of scope of this PP-Module. It only specifies the policy controlling the TOEs response to indicators from those processing components or to take advantage of the requested server's subject attributes exposed by decryption

FDP_PPP_EXT.1.4

The TSF shall allow the Plaintext Processing Policy to be applied at each information flow control point between inspection processing functional components, including any network interface used to support external inspection processing.

Application Note: This element indicates where the TLS plaintext processing policy can be assigned. A conforming TOE must be able to assign processing rules to prevent TLS data from being exposed to unauthorized processing units based on the requested server attributes, and to allow TLS sessions containing malicious or unauthorized data, as determined by the inspection processing functional components, to be blocked at the earliest possible point, avoiding compromise of the TOE or the monitored client.

FDP_PPP_EXT.1.5

The TSF shall

- drop Information flows between inspection processing components, including any interface to external inspection processing components, that cannot be associated to an existing TLS session thread.
- inform the TLS session establishment policy of the TLS session thread associated to any information flow that is blocked by the plaintext processing policy.

Application Note: This element identifies state information shared by the TLS inspection processing policy and the TLS session establishment policy associated. The TSF may inform the TLS session establishment policy that it has blocked a data flow either explicitly, by sharing state information, signaling, or other mechanism, or implicitly via the use of time-out mechanisms.

Evaluation Activities

FDP_PPP_EXT.1
TSS
The evaluator shall examine the TSS to validate that internal routing functions or controls associated with the Plaintext Processing Policy are described.
Guidance
The evaluator shall inspect the operational guidance documents and ensure that instructions for any configurable features of the Plaintext Processing Policy function are provided.
Tests
The evaluator shall perform the following tests:

- ***Test 1:*** *For each routing option described in the routing policy, the evaluator shall attempt to construct a data flow that exercises the routing option and observe the intended routing occurs.*
- ***Test 2:*** *For each routing option described in the routing policy, the evaluator shall attempt to construct a data flow that violates the routing option and observe that the violation is detected and the flow blocked.*

FDP_PRC_EXT.1 Plaintext Routing Control

FDP_PRC_EXT.1.1

The TSF shall control the routing of information flows containing plaintext within a TLS session thread in accordance with the configured Plaintext Processing Policy identified in FDP_PPP_EXT.1.

FDP_PRC_EXT.1.2

The TSF shall separate information flows containing plaintext within different TLS session threads.

FDP_PRC_EXT.1.3

The TSF shall not expose plaintext within a TLS session thread except to

inspection processing functional components identified in, and as authorized by the configured Plaintext Processing Policy, as described in FDP_PPP_EXT.1.

Evaluation Activities

FDP_PRC_EXT.1
TSS
The evaluator shall examine the TSS to validate that each interface between inspection processing functional components and TLS decryption/encryption buffers that can be used to control the routing of decrypted plaintext associated to a TLS session thread and the internal routing events or rules that control internal routing of decrypted plaintext at each interface are described.
Guidance
The evaluator shall inspect the operational guidance documents and ensure that instructions for any configurable features of the Plaintext Routing function are provided.
Tests
The evaluator shall perform the following tests:

- ***Test 1:*** *The evaluator shall configure the TSF and establish monitored clients and requested servers to establish multiple TLS session threads through the inspection processing functional components, in which the plaintext in each thread is distinguishable, either by the expected response of an inspection processing functional component, or by logs. The evaluator shall examine the observable responses and logs to confirm that the threads are treated separately.*
- ***Test 2:*** *(conditional, the TSF can establish plaintext processing rules that exclude plaintext processing by a particular inspection processing component): The evaluator shall configure the TSF, configure a plaintext processing policy, and establish monitored clients and requested servers to establish a TLS session thread through the inspection processing functional components for which the configured plaintext processing rules prohibits the processing of the data by a particular inspection processing component. The evaluator shall examine the logs and/or inspection processing response to determine that data is not processed by the component.*

FDP_RIP.1 Subset Residual Information Protection

FDP_RIP.1.1

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*] the following objects: [**assignment**: *list of objects*].

Application Note: “Resources” in the context of this requirement are any data buffers used to implement STIP functions, including the TLS buffers containing decrypted TLS payloads. The concern is that a buffer or memory area might be reused in subsequent function or communication channel resulting in inappropriate disclosure of sensitive data. “Objects” refers to any sensitive data objects that are under control of the TSF.

Evaluation Activities

FDP_RIP.1
TSS
The evaluator shall examine the TSS to ensure that, at a minimum, it describes how the previous information content is made unavailable, and at what point in the buffer processing this occurs.
Guidance
There are no guidance EAs for this component.
Tests
There are no test EAs for this component.

FDP_STG_EXT.1 Certificate Data Storage

FDP_STG_EXT.1.1

The TSF shall use [**selection**: *access controlled storage*, *an integrity mechanism*] to protect the trusted public keys and certificates (trust store elements) used to validate local logon, trusted channel, and external communication to the STIP.

Application Note: If “an integrity mechanism” is selected, FCS_CKM_EXT.5 must be included in the ST

Evaluation Activities

FDP_STG_EXT.1
TSS
The evaluator shall examine the TSS to ensure it describes the trusted public keys and certificates implemented, including trust stores that contain root CA certificates used to meet the requirements of this PP. This description shall contain information pertaining to how certificates are loaded into the store, and (if the first selection in this SFR is selected) how the store is protected from unauthorized access in accordance with the permissions established in FMT_SMF.1 and FMT_MOF.1.
Guidance
The evaluator shall examine the operational guidance to ensure it contains instructions for how to load certificates and public keys into, and remove certificates and public keys from the protected storage or apply (trust) or remove (untrust) the indicated protection mechanism.
Tests
This test is conditional on the first option in the selection of this SFR being chosen. If the second option is chosen, the evaluator does not perform this and instead performs the actions called for in FCS_CKM_EXT.5.

The evaluator shall attempt to modify the contents of the Trust Anchor Database in a way that violates the documented permissions and verify that the attempt fails.

FDP_STIP_EXT.1 SSL/TLS Inspection Proxy Functions

FDP_STIP_EXT.1.1

The TSF shall be capable of performing the Inspection Operation consisting of establishing a TLS session between TOE and the requested server according to FCS_TTTC_EXT.1, establishing a TLS session between the monitored client and the TOE according to FCS_TTTS_EXT.1, and routing decrypted application data from either of these TLS sessions to or between inspection processing functional components within the TOE, or between the TOE and external inspection processing functional components to a unique TLS session thread, according to FDP_PPP_EXT.1 and FDP_PRC_EXT.1.

Application Note: This defines the inspection operation, where the TLS connection is terminated at both ends on the TOE and the opportunity for inspection of the contents is allowed.

FDP_STIP_EXT.1.2

The TSF shall obtain a certificate from the TOE CA that represents the requested server for establishment of the TLS session with the monitored client when performing an inspect operation.

Application Note: Certificates are generated by the TOE's embedded CA function, or obtained from an optional certificate cache maintained by the TOE. Certificate caching is not required, however, in the case where certificate caching is supported, the TSF will still need to perform certificate generation if no corresponding cache entry can be found for the requested server that matches the current certificate profile.

FDP_STIP_EXT.1.3

The TSF shall [**selection**: *require administrator confirmation of consent, provide a consent to monitor banner to the client, in accordance with FTA_TAB.1**/TLS**, and receive an affirmative response*] prior to sending decrypted TLS application data from a monitored client to inspection processing functional component as part of an inspection operation.

Application Note: The selection "require administrator confirmation of consent" means that there is a means for the administrator to approve the operation based on receiving consent from the monitored client(s). This would include "real-time" approval mechanisms (a pop-up, for instance, accessible to an administrator) as well as configuration settings indicating "pre-approval" (again, only accessible by the administrator) such as one-time approval at installation (prior to any decryption), or included in a logon banner for administrators. A particular mechanism is not specified as it is up to the implementation. The intent is simply to ensure consent is obtained prior to monitoring.

FDP_STIP_EXT.1.4

The TSF shall provide the Bypass operation functionality by forwarding traffic between the monitored client and requested server such that monitored client can establish and maintain a TLS connection to the requested server.

Application Note: This merely defines the Bypass operation, where the STIP does not inspect the traffic, and just forwards packets between the monitored client interface and the requested server interface.

FDP_STIP_EXT.1.5

When initiating a Block operation, the TSF shall be capable of providing a [**selection**: *TLS error response*, *[**assignment**: other error message]*] to the monitored client associated with the blocked TLS session.

Application Note: This requires the TOE to provide some form of notification to monitored client when the monitored client attempts to initiate a connection and that connection is blocked. This can be done through the TLS error response, or (using the "assignment" part of the selection) some other means defined by the ST author.

Evaluation Activities

FDP_STIP_EXT.1.1
TSS
The evaluator shall examine the TSS to ensure that inspection operation is described.

The evaluator shall examine the TSS to ensure that the logical components of a TLS session thread are described, and that a method for tracking the data flows associated to a TLS session is described. The evaluator shall check the TSS and verify that all components of a TLS session thread are included in the TLS session thread description. The evaluator shall examine the TSS to ensure that separation mechanisms between TLS session threads is described. If TLS resumption is supported, as indicated by the final selection in FCS_TTTC_EXT.1, and/or the final selection in FCS_TTTS_EXT.1, the evaluator shall examine the TSS and verify that the description explains how TLS resumption does not create TLS sessions that are included in multiple TLS session threads.
Guidance
The evaluator shall examine the operational guidance to ensure instructions for any configurable features of the inspection operation, and any configurable features of the TLS session thread management to meet the requirements are provided.
Tests
The evaluator shall follow the operational guidance to configure the TSF. The evaluator shall establish two monitored clients (client a, and b) able to initiate a TLS session that is compliant with FCS_TLSC_EXT.1, and two servers (servers 1 and 2) able to establish TLS sessions in accordance with FCS_TLSS_EXT.1, each of which which has certificate that is valid in accordance with FIA_X509_EXT.1/STIP, and which is issued by a CA, different than the TOE's embedded CA, that is trusted by the TOE. If the TSF supports TLS resumption, the evaluator shall configure server 1 to support TLS resumption using a mechanism (tickets or session number) supported by the TSF, and configure server 2 to refuse TLS resumption and instead respond with a full TLS handshake when requested to do resumption. The evaluator shall follow AGD guidance to configure the TLS session establishment policy so TLS sessions through the TOE between each of the client-server combinations will be inspected. The evaluator shall use appropriate tools to monitor the traffic between the clients and the TOE, and between the TOE and the server to observe the TLS handshake messages. If the TSF supports TLS session resumption, the evaluator shall clear any TLS session state that might be retained by the TSF and configure the TSF to use session resumption. Note that tests 1 and 2 have additional instructions if TLS resumption is supported, but apply regardless of TLS resumption support. Test 3 should only be performed if TLS resumption is supported. The evaluator shall perform the following tests, in order:

- ***Test 1:*** *The evaluator shall initiate a TLS session from client 'a' to server 1, and initiate a TLS session from client 'b' to server 2. The evaluator shall observe the traffic between the TOE and the servers the data decrypted at the servers to verify that the TLS sessions are distinct. The evaluator shall also observe the traffic between the clients and the TOE and observe that the TLS sessions are distinct. The evaluator shall note and retain the TLS session information for the remaining tests and ensure that the sessions are not terminated during Test 2.*
- ***Test 2:*** *The evaluator shall retain the state of the TOE from Test 1. If TLS resumption is supported, the evaluator shall ensure the TLS state in server 1 is retained. The evaluator shall initiate a TLS session from client 'b' to server 1 through the TOE. The evaluator shall observe the traffic between the TOE and server 1, and data received at server 1 to confirm the TLS session thread between client 'b' and server 1 is different than the TLS session between the TOE and server 1 associated to the TLS session thread between client 'a' and server 1 established in Test 1. The evaluator shall observe that the TLS session between client 'b' and the TOE associated to the TLS session thread between client 'b' and server 1 is different than the TLS session between client 'b' and the TOE associated to the TLS session thread between client b and server 2 established in test 1. The evaluator shall terminate the TLS sessions from client 'b' to the TOE and observe that both TLS sessions associated to the TLS session threads, one to server 1 and the other to server 2, are terminated.*
- ***Test 3:*** *(conditional, the TSF supports session resumption): The evaluator shall initiate a TLS session resumption between client 'b' and server 2 through the TOE, and observe that the TSF responds with a full TLS handshake.*

FDP_STIP_EXT.1.2

TSS

The evaluator shall examine the TSS to ensure it contains a description of the TOEs embedded certification authority function and any certificate caching in support of the inspection operation.

Guidance

The evaluator shall examine the operational guidance to ensure instructions to configure the TOE's embedded CA function and any certificate caching function required to meet the requirements is provided.

Tests

The evaluator shall perform the following tests:

- ***Test 1:*** *The evaluator shall configure and establish a monitored client and a requested server, and ensure the requested server has a certificate issued by a CA trusted by the TSF, but different than the TOE's embedded CA. The requested server certificate shall contain a valid identifier of DNS name type identifying the requested server subject alternate name extension. The monitored client will be configured to send the same DNS name for the requested server in the SNI extension of its Client Hello. The evaluator shall follow AGD guidance to configure the TLS session establishment policy to inspect TLS sessions between the monitored client and the requested server. The evaluator shall initiate a TLS session between the monitored client and the requested server through the TOE, and observe that the certificate received in the server certificate message at the monitored client is issued by the TOE's embedded signing certificate and contains the same DNS name for the server in the subject alternate name extension, as requested by the client.*
- ***Test 2:*** *(conditional, the TSF supports certificate caching): The evaluator shall follow the operational guidance to configure the TSF to retain generated certificates in cache for a short time. The evaluator shall establish three monitored clients and a single requested server, and follow AGD guidance to ensure TLS sessions between the monitored clients and the requested server are inspected as in Test 1. The evaluator shall establish a TLS session from two of the monitored clients to the same requested server within the configured cache time, and confirm that the certificates received at each client are identical. The evaluator shall wait until the cache time has expired, and then initiate a TLS connection from the third monitored client and note that the certificate received at the third client is different than the previous certificates receive at the first two clients.*

FDP_STIP_EXT.1.3

TSS

The evaluator shall examine the TSS to ensure it describes the mechanism used to determine clients have consented to monitoring in accordance with the requirement. If the second option in the selection is claimed, the evaluator shall confirm that the TSS includes a description of the confirmation exchange between the TSF and monitored clients.

Guidance

The evaluator shall examine the operational guidance to confirm that any instructions to configure the TSF to meet this requirement are provided. If the second option in the selection is claimed, the evaluator shall confirm that instructions for configuring the consent banner is provided.

Tests

The following test is conditional on the TSF supporting a consent to monitor banner for monitored clients:

The evaluator shall establish a monitored client and requested server, and follow AGD guidance to configure the TSF to present monitored clients a consent to monitor banner. The evaluators shall follow AGD guidance to inspect TLS traffic between the monitored client and requested server, and initiate a TLS session from the monitored client to the requested server through the TOE. The evaluator shall observe that the consent to monitor banner is provided to the client and that no traffic from the client is inspected until consent is provided.

FDP_STIP_EXT.1.4

TSS

The evaluator shall examine the TSS to ensure that the Bypass Operation is described.

Guidance

The evaluator shall examine the operational guidance to verify that instructions for configuring the bypass operation, to include logging of bypassed TLS sessions, is provided.

Tests

The evaluator shall establish a monitored client and a requested server. The evaluator shall follow AGD guidance to configure the TOE and its TLS session establishment policy so that TLS traffic between the monitored client and the requested server is processed via the Bypass Operation and so that bypassed TLS sessions are logged. The evaluator shall initiate a TLS session from the monitored client to the requested server through the TOE. The evaluator shall monitor traffic between the monitored client and the TOE and between the TOE and the requested server. The evaluator shall then observe that the TLS client handshake messages between the client and the TOE are identical to the client handshake messages between the TOE and the server, and that the TLS server handshake messages between the server and the TOE are identical to the TLS server handshake messages between the TOE and the client. The evaluator shall observe the TOE logs to ensure that the TLS session between the client and server is logged.

FDP_STIP_EXT.1.5
TSS
The evaluator shall examine the TSS to ensure that the block operation is described and includes the response to the monitored client when TLS sessions are blocked.

The evaluator shall examine the TSS to ensure that all events that initiate a transition to the block operation are described.
Guidance
The evaluator shall examine operational documentation and verify that instructions to configure any configurable features of the block operation are provided.
Tests
The evaluator shall establish a monitored client and a requested server. The evaluator shall follow AGD guidance to configure the TOE and its TLS session establishment policy so that TLS sessions between the monitored client and the requested server through the TOE are processed by the block operation. The evaluator shall initiate a TLS session from the monitored client to the requested server through the TOE and observe that the TLS session is blocked. The evaluator shall confirm that the monitored client receives the specified error message.

FDP_TEP_EXT.1 SSL/TLS Inspection Proxy Policy

FDP_TEP_EXT.1.1

The TSF shall perform SSL/TLS Inspection Proxy functions and enforce SSL/TLS Inspection Proxy rules on TLS traffic received by the TSF from monitored clients and servers requested by monitored clients, and on TLS traffic controlled by the TSF to be sent to monitored clients and servers requested by monitored clients.

Application Note: This element defines the policy and requires the rules (defined in other elements of this component) to be applied to TLS network traffic from monitored clients to requested servers that is processed at the TOE's network interfaces (as required in subsequent elements).

This requirement is to be enforced even if the network interfaces are saturated/overwhelmed with network traffic.

The requirement only applies to network traffic at the external interfaces that is identified as TLS traffic between a monitored client and requested server. This does not apply, for instance, to TLS traffic associated with administration of the STIP.

FDP_TEP_EXT.1.2

The TSF shall allow the definition of SSL/TLS Inspection Proxy rules based on the following attributes of each monitored client and requested server: [

- *Network Protocol fields: [**selection**: IPv4, IPv6, [**assignment**: other internet protocol]]:*
  - *Source address*
  - *Destination address*
  - *Source port*
  - *Destination port*
  - *[**selection**: [**assignment**: other fields containing identity attributes for the monitored client or requested server], no other fields]*
- *TLS Client Hello handshake message:*
  - *Server_name extension of the requested server*
  - *Client side interface*
- *TLS Server Certificate message:*
  - *Issuer*
  - *Subject*
  - *SubjectAlternateName*
- *Distinct interface:*
- *[**selection**:*
  - *TLS client certificate message [**selection**:*
    - *Certificate issuer,*
    - *Certificate subject,*
    - *Certificate subject alt name*

    ] ,
  - *[**assignment**: other attributes],*
  - *no other attributes*

  ]

].

Application Note: This element identifies the various attributes that are applicable when constructing rules to be enforced by this requirement. The rules

apply to external interfaces receiving TLS messages from a monitored client (client side interface), and to the TLS messages received by the TOE in response to the TSF initiating a TLS connection to a server requested by the monitored client (server side interface).

Network Protocol fields are used by the TSF to determine the IP address of the monitored client and the IP address of the requested server in traffic received on the client side interface only. Indicate which network protocols, including the internet layer and transport layer protocols that are used to determine the indicated fields that can be applicable when constructing rules for this policy.

The TLS Client Hello messages, and optional client certificate messages are received on the client side interface only. If the TSF supports client authentication, 'TLS client certificate message' should be selected (with the appropriate subselections supported by the TOE), and FCS_TTTS_EXT.3, Authentication of Monitored Clients should be claimed.

The TLS certificate message is received on a server side interface prior to the TSF sending a Server Hello done message on the client side interface.

FDP_TEP_EXT.1.3

The TSF shall allow the following operations to be associated with SSL/TLS Inspection Proxy rules: block, bypass, or inspect, with the capability to log the operation.

FDP_TEP_EXT.1.4

The TSF shall be able to define monitored clients, requested servers, and [**selection**: *specific client-server connections*, *no other abstractions*] in terms of the attributes associated with the SSL/TLS Inspection Proxy function.

Application Note: This element requires that there must be a mechanism to define a "monitored client" and "requested server" via the attributes specified in FDP_TEP_EXT.1.2. This entity will then have associated rules defined in other elements in this component related to the STIP functionality and operations. If the TOE is able to define a set of attributes that represent a unique client-server connection, then the first selection item should be chosen.

FDP_TEP_EXT.1.5

The TSF shall be able to associate a monitored client, requested server, and [selection: specific client-server connections, no other abstractions] with the allowed TLS version or versions, TLS cipher suites (including TLS key exchange algorithms and key sizes), the supported groups per FCS_TTTC_EXT.5.1, and **[selection: *mutual authentication block-bypass, requested server certificate revocation status unavailable, critical extension in a certificate unrecognized, nothing else*]** that shall be used when performing the SSL/TLS Inspection Proxy operations.

Application Note: This element requires a mechanism that defines, for each "monitored client" and "requested server" (and, if supported, unique client-server pairs), the allowed set of the indicated TLS characteristics associated with those entities. This association allows the enforcement of rules defined by other elements in this component. The first selection is chosen if the TOE supports rules based on both a monitored client and requested server pair.

The second selection indicates events specified in other requirements that need to be associated with monitored clients/requested servers in rules so that appropriate actions can be taken.

The first item is chosen if the TOE supports multiple responses to a client certificate request message from a requested server; FDP_TEP_EXT.1.7 and its Application Note have additional details.

The second item is chosen if FIA_X509_EXT.2 indicates a privileged user may indicate their choice on whether to accept a requested server certificate for which revocation information is not available using allowances. See also FDP_TEP_EXT.1.8.

The third item is chosen if the TOE supports detection of a critical extension in a certificate (being validated according to FIA_X509_EXT.1/STIP) that it cannot interpret. RFC 5280 indicates that this situation results in an invalid certificate, but FIA_X509_EXT.1/STIP provides an additional option that—instead of treating the certificate as invalid (and thus blocking the connection)—the administrator can indicate that the "Bypass" operation is to be applied to the connection instead (which essentially defers the decision to make the connection to the client). See also FDP_TEP_EXT.1.8.

FDP_TEP_EXT.1.6

The TSF shall allow the SSL/TLS Inspection Proxy rules to be assigned to each distinct network interface.

FDP_TEP_EXT.1.7

The TSF shall [**selection**:

- *perform a [**selection**: block, bypass, mutual authentication inspection] operation ,*
- *send an empty certificate list as part of the inspection operation*

] on the session when receiving a TLS certificate request message from the requested server when establishing the TLS in accordance with FCS_TTTC_EXT.1.

Application Note: The ST author will select one or more response options according to the capabilities of the TSF. A mutual authentication inspection operation is a variant of the inspection operation. If this item is selected, the mutual authentication SFRs FCS_TTTC_EXT.3 and FCS_TTTS_EXT.3 must be claimed. If mutual authentication is not supported, one or more of the remaining options is selected: 'Block' and 'send an empty certificate list as part of the inspection operation' are alternative methods to ensure that certificates issued by the TOE's embedded certificate authority are not provided to requested servers that are not known to trust the CA. Block is initiated by the TSF; the TOE terminates the TLS session, whereas 'send an empty certificate list...' allows the requested server to continue with the TLS session without client authentication or terminate the session.

Inspection of mutual authenticated TLS requires both the client and server to trust the embedded CA, and therefore has limited use. It is preferred that inspection of mutual authenticated TLS be performed by components of the requested server security architecture (e.g. via a traffic filtering firewall or an attribute-based access control mechanism) and not be performed by devices described in this PP-Module. If mutual authentication inspection is selected, then the selection-based requirements FCS_TTTC_EXT.3 and FCS_TTTS_EXT.3 will be included by the ST author, and the “mutual authentication” item will be selected in FCS_TTTC_EXT.1.1 and FCS_TTTS_EXT.1.1.

If more than one response option is selected, the ‘mutual authentication block-bypass’ exception specification must be claimed in FDP_TEP_EXT.1.5 and be configurable within the TLS session establishment policy to determine which of the supported operations will be applied for a specific requested server. It is expected, but not required, that one of the selected operations will be a default operation and the other determined by the server matching the exception specification.

FDP_TEP_EXT.1.8

The TSF shall

- Block the connection if the monitored client does not support a TLS version, cipher suite, key exchange, and key size that are in its allowed set as defined in FDP_TEP_EXT.1.5
- Block the connection if the requested server does not negotiate a TLS version, cipher suite, key exchange, and key size that are in its allowed set as defined in FDP_TEP_EXT.1.5
- Either block or [**selection**: *require administrative approval to inspect or bypass*, *no other rule*] the connection if the requested server does not negotiate a TLS version, cipher suite, key exchange, and key size that are in the set proposed by the monitored client in its Client Hello message
- Block or [**selection**: *inspect*, *bypass*, *no other rule*] the connection if TOE certificate processing indicates revocation information is not available for a requested server or [**selection**: *monitored client*, *no other entity*]
- Block or [**selection**: *bypass*, *no other rule*] a connection if TOE certificate processing indicates an uninterpretable critical extension is present in the certificate of a requested server.

Application Note: Support by a client for the revocation information unavailable case is determined by the TLS handshake protocol messages and fatal errors from the client received during TLS session negotiation with the TOE in accordance with FCS_TTTS_EXT.1.

In the case where a critical extension is encountered that cannot be interpreted by the TOE in accordance with FIA_X509_EXT.1.1/STIP “bypass” can be selected in the last bullet item above. Note that it is not allowed for the administrator to select “Inspect” in this case.

FDP_TEP_EXT.1.9

The TSF shall enforce the following default SSL/TLS Inspection Proxy rules on all SSL/TLS network traffic received from interfaces associated with monitored clients and requested servers:

- The TSF shall drop and be capable of [**selection**: *counting*, *logging*] invalid TLS messages
- The TSF shall drop and be capable of logging TLS Client Hello messages for which no valid client can be determined
- The TSF shall drop a TLS Client Hello message for which no valid server attribute can be determined
- The TSF shall drop and be capable of [**selection**: *counting*, *logging*] TLS messages other than a Client Hello if the message is not associated with an existing TLS session thread established via the inspection operation or a TLS encrypted data flow established via a bypass operation
- The TSF shall terminate a TLS session thread if it receives a fatal TLS error message from the monitored client
- The TSF shall attempt to **[selection:**
  - ***resume the session,***
  - ***renegotiate the session,***
  - ***terminate the TLS session thread and provide a [selection: TLS error message, [assignment: method of notification] to the monitored client associated with the TLS session thread]***

  ] if it receives a fatal TLS error message on the TLS session to the requested server
- The TSF shall terminate a TLS session thread established via the inspect operation, and terminate a TLS encrypted data flow established by the bypass operation, if the TSF receives no traffic from the associated monitored client for a configurable period
- The TSF shall transition a TLS session thread state from inspect operation to block operation, when indicated to do so by the TLS plaintext processing policy

.

Application Note: Dropping a message, performing a block operation, and transitioning to a block operation are different. Dropping a message is typically a silent operation; performing block operation may require messages to be sent to the monitored client associated to the TLS Client Hello; transitioning to a block operation involves termination of the TLS session thread, and potentially sending TLS alert messages to the requested server and TLS alert messages or other messages to the monitored client.

FDP_TEP_EXT.1.10

The TSF shall block all connections for which an Inspection or Bypass operation is not defined.

Application Note: This is the deny by default rule. Note that the block rule does not need to be explicitly defined. This element should not be interpreted that all Client Hello packets should be blocked; the intent is that the Client Hello is initiated from the monitored client, and then the TOE performs processing to determine what to do with the requested connection. If it cannot find a rule that applies for the requested connection, then this element requires that the connection be blocked.

Evaluation Activities

FDP_TEP_EXT.1
TSS
The evaluator shall examine the TSS and verify that the TLS session establishment policy is adequately described. The evaluator shall verify that the TSS description of the TLS session establishment policy includes a discussion of the TOE's initialization/startup process, which clearly indicates where processing of TLS messages begins and provides a discussion that supports the assertion that TLS messages are dropped during this process.

The evaluator shall verify that the TSS also includes a narrative that identifies the components involved in processing TLS messages and describe the safeguards that would prevent inspection or Bypass Operation functions being performed in the event of a component failure. This could include the failure of a component or a failure within a component. The evaluator shall also verity that the TSS description indicates how the TLS protocol is recognized at each client side and server side interface.

The evaluator shall examine the TSS and verify that it describes any non-configurable rules implementing the TLS session establishment policy and that it describes how such rules invoke the inspect, bypass, or block operations based on the subject attributes included in FDP_TEP_EXT.1.2.

The evaluator shall verify that the TSS describes a TLS session establishment policy and the attributes identified in FDP_TEP_EXT.1.2 are identified as being configurable within the TLS session establishment policy rules. The evaluator shall verify that each configurable rule of the TLS session establishment policy can identify the block, bypass or inspect operation, with the option to log block and bypass operation.

The evaluator shall examine the TSS and verify that rules to define server allowances, client allowances, and other entity allowances (if supported) for TLS parameter usage and TLS processing errors that depend on the TLS session establishment policy is described and includes all conditions indicated in FDP_TEP_EXT.1.5. If multiple response options for receiving a client certificate request message from a requested server are selected in FDP_TEP_EXT.1.7, the evaluator shall confirm that the 'mutual authentication block-bypass' specification is claimed in FDP_TEP_EXT.1.5 and that a description of the processing rules for a TLS client certificate request are included in the TSS description of the TLS session establishment policy.

If mutual authentication for through-traffic processing is supported, the evaluator shall examine the TSS and verify that policy rules to define when mutual authentication is allowed are described.

The evaluator shall examine the TSS and verify that description of the TLS protocol and TLS session establishment policy describe the policy-specified behavior that results from TLS protocol errors as required in FDP_TEP_EXT.1.8.

The evaluator shall examine the TSS and verify that the default rules indicated in FDP_TEP_EXT.1.9 and FDP_TEP_EXT.1.10 are described.

Guidance

The evaluator shall examine the operational guidance to verify that instructions to configure the TLS session establishment policy are provided.

The evaluator shall examine the AGD guidance documents and verify that they identify all attributes included in FDP_TEP_EXT.1.2 as being configurable within the TLS session establishment policy, which is that all configurable features of the TLS session establishment policy function are described in the operational guidance.

The evaluator shall examine the AGD guidance documents and verify they indicate each rule can identify the following operations: block, bypass, and inspect. The evaluator shall confirm that instructions for configuring the inspection, bypass, and block operations within rules are included.

The evaluator shall examine the AGD guidance documents and verify they specify each rule indicating block or bypass operations can designate whether logging or counting of TLS Client Hello messages invoking the operation is performed.

The evaluator shall examine the AGD guidance documents and verify they provide instructions on configuring the TLS parameter allowances identified in FDP_TEP_EXT.1.5 and those responses to TLS protocol errors identified in FDP_TEP_EXT.1.8.are indicated.

The evaluator shall examine the AGD guidance documents and verify that any instructions required to configure the TLS session establishment policy to meet the requirements in this component are provided.

Tests

Setup: *The evaluator shall configure one or more monitored clients to present TLS requests to various TLS servers through the TOE. The TLS servers will obtain certificates issued by an external certification authority trusted by the TSF. The client, server, and the server certificates will meet the conditions described in each test. The evaluator shall configure the TOE according to operational guidance to have non-trivial rules for all TLS session establishment policy states. The evaluator shall conduct the following tests, establishing any additional configuration requirements as indicated in each.*

- ***Test 1:*** *For each rule of the TLS establishment policy indicating inspection operation processing, the evaluator shall ensure the monitored client is configured to meet the requirements for FCS_TLSC_EXT.1, the requested server is configured to meet the requirements for FCS_TLSS_EXT.1, and the server certificate is valid according to FIA_X509_EXT.1/STIP. The evaluator shall configure the TSF so the rule applies to the monitored client and requested server. The evaluator shall establish a TLS session from the monitored client to the requested server through the TOE. The evaluator shall then observe the TSF audit record, certificate repository, TLS Server Hello data received at the client, plaintext encrypted at the client, and plaintext decrypted by the requested server. The evaluator shall then confirm that the TSF established a TLS session with the requested server, issued a certificate representing the requested server, established a TLS session with the monitored client, decrypted the data, performed any inspection processing, and presented the data to the requested server via the established TLS session.*
- ***Test 2:*** *For each rule of the TLS establishment policy indicating bypass processing, the evaluator shall establish a monitored client, requested server, and server certificate that meets the rule. The evaluator shall send a TLS request from the monitored client to the requested server through the TOE, and then inspect logs, certificate repository, certificate received by the monitored client in the Server Hello message, plaintext encrypted by the monitored client, and plaintext decrypted by the requested server to confirm that bypass processing occurred.*

- ***Test 3:*** *The evaluator shall follow AGD guidance to ensure the TSF is configured to log blocked TLS sessions. For each rule of the TLS establishment policy indicating blocking of the TLS session, as indicated in any element of this component, the evaluator shall establish that a monitored client, a requested server, and a server certificate meet the rule. The evaluator shall send a TLS session from the monitored client to the requested server through the TOE and observe that the monitored client receives an error response in accordance with FDP_STIP_EXT.1.5 indicating that the session was blocked. The evaluator shall inspect the TSF logs to verify that each session was recorded as blocked.*
- ***Test 4:*** *For each event that initiates a transition from the inspection operation to the block operation, the evaluator shall attempt to establish a monitored client and requested server, and configure the TOE and its TLS session establishment policy to invoke the event. For each such event, the evaluator shall initiate a TLS session from the monitored client to the requested server through the TOE. The evaluator shall monitor traffic between the monitored client and the TOE, and monitor traffic between the TOE and the requested client, observing that TLS handshake messages prior to the event are sent, and that any TLS sessions established prior to the event are terminated on transition of the session to the block operation. The evaluator shall observe that the monitored client receives the specified error message indicating that the TLS session is blocked.*
- ***Test 5:*** *Test 5 [conditional, both 'mutual authentication inspection' and 'send an empty certificate list as part of the inspection operation' are selected in FDP_TEP_EXT.1.7]: The evaluator shall establish a server to send certificate requests in its TLS handshake. The evaluator shall exablish a monitored client configured to provide a valid client certificate in response to a certificate request. The evaluator shall follow AGD guidance to configure the TLS inspection proxy policy to send an empty certificate list in a certificate message to the server, and initiate a TLS request from a monitored client to the server through the TOE. The evaluator shall observe network traffic between the TOE and the requested server and confirm that the TOE sends an empty certificate list to the server after receiving the certificate request.*

  Using the same server, the evaluator shall follow AGD guidance to configure the TSF to perform mutual authentication inspection with the server, and initiate a TLS request from the same monitored client ro the same requested server through the TOE. The evaluator shall observe network traffic between the TOE and the requested server and confirm the TOE sends a certificate message containing a client certificate representing the monitored client.

5.2.5 Identification and Authentication (FIA)

FIA_ENR_EXT.1 Certificate Enrollment

FIA_ENR_EXT.1

The TSF shall be able to generate a certificate request to an external certification authority to receive a certificate for the TOE's embedded CA's signing key using [**selection**:

- *PKCS#10 in accordance with FIA_X509_EXT.3*,
- *Enrollment over Secure Transport (EST) in accordance with FIA_ESTC_EXT.1*

].

Application Note: The external certification authority may be a root or intermediate certification authority that is used to issue and manage the TOE's embedded CA's certificate. It is not to be used to directly issue end entity certificates to requested servers instead of the TOE's embedded CA.

Evaluation Activities

FIA_ENR_EXT.1
TSS
The evaluator shall examine the TSS to ensure that it describes the certificate enrollment function options.
Guidance
The evaluator shall examine the operational guidance documentation and confirm that it contains instructions for obtaining a certificate for the embedded CA using the options claimed in FIA_ENR_EXT.1.1.
Tests
Testing for this SFR is addressed through evaluation of FIA_X509_EXT.3 or FIA_ESTC_EXT.1, depending on the selections made in FIA_ENR_EXT.1.1.

FIA_X509_EXT.1/STIP X.509 Certificate Validation (STIP)

FIA_X509_EXT.1.1/STIP

The TSF shall validate certificates used for connections supporting STIP functions in accordance with the following rules:

- RFC 5280 certificate validation and certification path validation supporting a minimum path length of three certificates.
- The certification path must terminate with a trusted CA certificate designated as a trust anchor.
- The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.
- The TSF shall validate the revocation status of the certificate using [**selection**: *the Online Certificate Status Protocol (OCSP) as specified in RFC 6960, a Certificate Revocation List (CRL) as specified in RFC 5280 Section 6.3, Certificate Revocation List (CRL) as specified in RFC 5759 Section 5*].
- The TSF shall validate the extendedKeyUsage field according to the following rules depending on the certificate type and purpose:
  - Server certificates presented in a TLS certificate message for ThruTraffic processing TLS shall have meet one of the following checks:
    - There is no extendedKeyUsage field
    - The extendedKeyUsage field is present and contains the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1)
    - The extendedKeyUsage field is present and contains the ‘any’ purpose (id-…)
  - Server certificates presented for TLS not associated with the ThruTraffic processing include an extendedKeyUsage field that contains the ServerAuthentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1).
  - Code-signing certificates include the extendedKeyUsage field that contains the CodeSigning purpose.
  - Client certificates presented for TLS for any purpose shall include the extendedKeyUsage field that contains the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.
  - OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.
  - All other certificates used for any other purpose include an extendedKeyUsage field that DOES NOT contain the ‘any’ purpose.
- The TSF shall validate all extensions marked as critical and verify the value is appropriate for the functionality that uses the value.

FIA_X509_EXT.1.2/STIP

The TSF shall only treat a certificate as a CA certificate if the basicConstraints extension is present and the CA flag is set to TRUE.

Application Note: FIA_X509_EXT.1.1/STIP lists the rules for validating certificates for STIP Functions. The text that says what to do if revocation information is not available, or if a critical extension cannot be processed, is provided in FCS_TTTC_EXT.1.3.

The ST author selects whether revocation status is verified using OCSP or CRLs. The SFR indicates that the TOE be capable of supporting a minimum path length of three certificates. This means that the TOE supports a hierarchy comprising of at least a self-signed root CA certificate, a subordinate CA certificate, and a leaf certificate. The chain validation is expected to terminate with a trust anchor. This means the validation can terminate with any trusted CA certificate designated as a trust anchor. This CA certificate must be loaded into the trust store ('certificate store', ' trusted CA Key Store' or similar) managed by the TOE trust store. If the TOE’s trust store supports loading of multiple hierarchical CA certificates or certificate chains, the TOE must clearly indicate all certificates that it considers trust anchors. The validation of X.509v3 leaf certificates comprises several steps:

a. A Certificate Revocation Check refers to the process of determining the current revocation status of an otherwise structurally valid certificate. This must be performed every time a certificate is used for authentication. This check must be performed for each certificate in the chain up to, but not including, the trust anchor. This means that CA certificates that are not trust anchors, and leaf certificates in the chain, must be checked. It is not required to check the revocation status of any CA certificate designated a trust anchor, however if such check is performed it must be handled consistently with how other certificates are checked.
b. An expiration check must be performed. This check must be conducted for

each certificate in the chain, up to and including the trust anchor.

c. The continuity of the chain must be checked, showing that the signature on each certificate that is presented to the TOE is valid and the chain terminates at the trust anchor.
d. The presence of relevant extensions in each certificate in the chain such as the extendedKeyUsage parameters of the leaf certificate must correspond to SFR-relevant functionality. For example, a peer acting as a web server should have TLS Web Server Authentication listed as an extendedKeyUsage parameter of its X.509v3 certificate. The TOE ensuresthat the relevant extensions in each certificate in the chain such as the extendedKeyUsage parameters of the leaf certificate correspond to the SFR-relevant functionality they are used with.

It is expected that revocation checking is performed when a certificate is used in an authentication step. It is expected that revocation checking is performed on both leaf and intermediate CA certificates when a leaf certificate is presented to the TOE as part of the certificate chain during authentication. Revocation checking of any CA certificate designated a trust anchor is not required.

If the TOE implements mutual authentication or acts as a server, there is no expectation of performing any checks on TOE's own leaf certificate during authentication. FIA_X509_EXT.1.2/STIP applies to certificates that are used and processed by the TSF and restricts the certificates that may be added as trusted CA certificates.

Evaluation Activities

FIA_X509_EXT.1/STIP

TSS

The evaluator shall ensure the TSS describes where the check of validity of requested server TLS certificates, associated OCSP certificates, and if mutual authentication for through-traffic processing is supported, where the check of validity of monitored client TLS certificates takes place.

The TSS shall describe when revocation checking is performed and on what certificates. If the revocation checking during authentication is handled differently depending on whether a full certificate chain or only a leaf certificate is being presented, any differences must be summarized in the TSS section and explained in the guidance.

It is expected that revocation checking is performed when a certificate is used in an authentication step and on both leaf and intermediate CA certificates when a leaf certificate is presented to the TOE as part of the certificate chain during authentication. Revocation checking of any CA certificate designated a trust anchor is not required. It is not sufficient to perform a revocation check of a CA certificate that is not designated a trust anchor (e.g., for an intermediate CA), only when it is loaded onto the device.

Guidance

There are no guidance EAs for this component.

Tests

The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication of a requested server certificate, or, if mutual authentication for throughtraffic processing is supported, a monitored client certificate, as well as CA certificates included in the certificate path and any for OCSP responses used in validating these certificates. The evaluator shall perform the following tests for FIA_X509_EXT.1.1/STIP. These tests must be repeated for each distinct security function that uses X.509v3 certificates in association with through-traffic processing. For example, if the TOE implements mutual authentication for through-traffic processing, then it shall be tested with each of FCS_TTTC_EXT.1 and FCS_TTTS_EXT.3.

- ***Test 1:*** *The evaluator shall present the TOE with a valid chain of certificates (terminating in a trusted CA certificate) as needed to validate the leaf certificate to be used in the function, and shall use this chain to demonstrate the function succeeds. Test 1a shall be designed so that the chain can be broken in Test 2 by either being able to remove the trust anchor from the TOEs trust store or by setting up the trust store in a way that at least one intermediate CA certificate needs to be provided together with the leaf certificate from outside the TOE, to complete the chain (e.g. by storing only the root CA certificate in the trust store).*
- ***Test 2:*** *The evaluator shall then 'break' the chain used in Test 1a by either removing the trust anchor in the TOE's trust store used to terminate the chain, or by removing one of the intermediate CA certificates (provided together with the leaf certificate in Test 1) to complete the chain. The evaluator shall show that an attempt to validate this broken chain fails.*
- ***Test 3:*** *The evaluator shall demonstrate that validating an expired certificate results in the function failing.*
- ***Test 4:*** *The evaluator shall test that the TOE can properly handle revoked certificates–conditional on whether CRL or OCSP is selected; if both are selected, then a test shall be*

performed for each method. The evaluator shall test revocation of the peer certificate and revocation of the peer intermediate CA certificate (i.e. the intermediate CA certificate should be revoked by the root CA). The evaluator shall ensure that a valid certificate is used, and that the validation function succeeds. The evaluator shall then attempt the test with a certificate that has been revoked (for each method chosen in the selection) to ensure when the certificate is no longer valid that the validation function fails. Revocation checking is only applied to certificates not designated as trust anchors. Therefore, the revoked certificates used for testing shall not be a trust anchor.

- ***Test 5:*** *If OCSP is selected, the evaluator shall configure the OCSP server or use a man-in-the-middle tool to present a certificate that does not have the OCSP signing purpose and verify that validation of the OCSP response fails. If CRL is selected, the evaluator shall configure the CA to sign a CRL with a certificate that does not have the cRLSign key usage bit set, and verify that validation of the CRL fails.*
- ***Test 6:*** *The evaluator shall modify any byte in the first eight bytes of the certificate and demonstrate that the certificate fails to validate (the certificate will fail to parse correctly).*
- ***Test 7:*** *The evaluator shall modify any byte in the last byte of the certificate and demonstrate that the certificate fails to validate (the signature on the certificate will not validate).*
- ***Test 8:*** *The evaluator shall modify any byte in the public key of the certificate and demonstrate that the certificate fails to validate (the hash of the certificate will not validate).*

The evaluator shall perform the following tests for FIA_X509_EXT.1.2/STIP. The tests described must be performed in conjunction with the other certificate services assurance activities, including FCS_TTTC_EXT.1 and FCS_TTTS_EXT.3 if claimed. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules, where the TSS identifies any of the rules for extendedKeyUsage fields for through-traffic processing (in FIA_X509_EXT.1.1/STIP).

The goal of the following tests to verify the TOE accepts a certificate as a CA certificate only if it has been marked as a CA certificate by using BasicConstraints with the CA flag set to True (and implicitly tests that the TOE correctly parses the BasicConstraints extension as part of X509v3 certificate chain validation).

For each of the following tests the evaluator shall create a chain of at least three certificates: a self-signed root CA certificate, an intermediate CA certificate, and a leaf (node) certificate. The properties of the certificates in the chain are adjusted as described in each individual test below (and this modification shall be the only invalid aspect of the relevant certificate chain).

- ***Test 1:*** *The evaluator shall ensure that at least one of the CAs in the chain does not contain the BasicConstraints extension. The evaluator shall confirm that the TOE rejects such a certificate at one (or both) of the following points: (i) as part of the validation of the leaf certificate belonging to this chain or (ii) when attempting to add a CA certificate without the BasicConstraints extension to the TOE's trust store (i.e. when attempting to install the CA certificate as one which will be retrieved from the TOE itself when validating future certificate chains).*
- ***Test 2:*** *The evaluator shall ensure that at least one of the CA certificates in the chain has a BasicConstraints extension in which the CA flag is set to FALSE. The evaluator shall confirm that the TOE rejects such a certificate at one (or both) of the following points: (i) as part of the validation of the leaf certificate belonging to this chain or (ii) when attempting to add a CA certificate with the CA flag set to FALSE to the TOE's trust store (i.e. when attempting to install the CA certificate as one which will be retrieved from the TOE itself when validating future certificate chains).*

  The evaluator shall repeat these tests for each distinct use of certificates for through-traffic processing. For example, use of certificates for establishing a TLS connection to a requested server is distinct from use of certificates for client authentication of a monitored client, if supported, and both of these uses would be tested.

5.2.6 Security Management (FMT)

The TOE is not required to maintain a management interface for STIP functions that is separate from the management functions that the NDcPP requires. If some or all STIP functions are managed separately, the relevant management functions from the NDcPP also apply to any management interfaces used specifically to meet the requirements of this PP-Module.

FMT_MOF.1 Management of Functions Behavior

FMT_MOF.1.1

The TSF shall restrict the ability to [*modify the behaviour of*] the functions [**assignment**: *functions defined in the Management Functions and Privileges table that are claimed in FMT_SMF.1.1 and FMT_SMF.1.1/STIP*] to [**assignment**: *roles defined in the Management Functions and Privileges table*].

Management Function	Security Administrator	Auditor	Account Manager	CA Operations

				Staff
Base-PP Mandatory Management Functions (FMT_SMF.1 and FMT_SMF.1/STIP)				
Ability to administer the TOE locally and remotely	M	CM	CM	CM
Ability to configure the access banner	M	-	O	O
Ability to update the TOE and to verify the updates	M	-	O	O
Ability to configure the authentication failure parameters for FIA_AFL.1	M	-	O	O
Ability to manage user accounts	C	-	CM	-
Ability to manage remote audit mechanism	M	CM	-	-
Ability to perform on-demand integrity tests	O	O	O	O
Ability to import and remove X.509v3 certificates into/from the Trust Anchor database	C	-	-	CM
Ability to configure identifying information for the TOE's embedded CA	C	-	-	CM
Ability to configure a maximum certificate validity duration	C	-	-	CM
Ability to manage inspection policy	O	-	-	O
Ability to configure inspection processing details	O	-	-	O
Base-PP Selectable Management Functions (FMT_SMF.1 and FMT_SMF.1/STIP)				
Ability to start and stop services	O	-	-	O
Ability to configure local audit behavior	O	O	-	-
Ability to modify the behavior of the transmission of audit data to an external IT entity, the handling of audit data, the audit functionality when Local Audit Storage Space is full	M	CM	-	-
Ability to search local audit	C	CM	-	-
Ability to configure the list of TOE-provided services available before an entity is identified and authenticated, as	M	-	O	-

specified in FIA_UIA_EXT.1				
Ability to manage cryptographic keys	M	-	-	CM
Ability to configure the cryptographic functionality	M	-	-	O
Ability to configure thresholds for SSH rekeying	M	-	O	O
Ability to configure the lifetime for IPsec SAs	M	-	-	-
Ability to configure the interaction between TOE components	M	O	-	O
Ability to enable or disable automatic checking for updates or automatic updates	M	-	-	-
Ability to re-enable an administrator account	C	-	CM	-
Ability to set the time which is used for time-stamps	M	O	-	O
Ability to configure NTP	M	-	-	-
Ability to configure the reference identifier for the peer	M	-	-	O
Ability to manage the TOE's trust store and designate X.509v3 certificates as trust anchors	M	-	-	CM
Ability to import X.509v3 certificates to the TOE's trust store	M	-	-	CM
Ability to configure and manage certificate profiles	C	-	-	CM
Ability to revoke issued certificates	C	-	-	CM
Ability to configure certificate status services	C	-	-	CM
Ability to configure automated process used to approve the revocation of a certificate or information about the revocation of a certificate	C	-	-	CM
Ability to clear a cache of valid issued certificates	M	-	-	CM
Ability to configure rules for automated issuance of certificates	C	-	-	CM
Ability to modify the CRL and/or OCSP configuration	C	-	-	CM
Ability to import private	C	-	-	CM

keys				
Ability to configure the TOE's behavior on validating certificates whose revocation status cannot be determined	M	-	-	CM
Ability to configure the TOE's behavior when non-supported critical extensions occur in a requested server certificate	C	-	-	CM
Ability to generate and export PKCS#10 messages	C	-	-	CM
Ability to configure EST functionality to generate and export EST requests	C	-	-	CM
Ability to configure TLS error responses for monitored clients	M	-	-	O
Ability to configure notification and consent message for monitored clients	M	-	-	O
Ability to configure rules for displaying a notification and consent message for acknowledgement prior to TLS inspection processing	M	-	-	O
Ability to search the certificate repository	C	CM	-	CM

Application Note: The management functions defined in this table are the same as the management functions defined in FMT_SMF.1 in the Base-PP as well as those defined in FMT_SMF.1/STIP in this PP-Module. The NDcPP only requires management functionality to be performed by a Security Administrator. This PP-Module defines optional additional management roles that may be claimed; if any of these are selected, this PP-Module provides guidance on how to apply those roles to the Base-PP's management functions.

The Management Functions and Privileges table uses the following key: M: Mandatory (this role must perform this function)

CM: Conditionally Mandatory (if this role is part of the TSF it must perform this function but if the role itself does not exist it may be satisfied elsewhere)

C: Conditional (if a role does not exist to satisfy a conditionally mandatory function, this is that function’s “backup” role)

O: Optional (this role may or may not perform this function)

The ST author should reproduce this table and update as needed to show which functions are implemented by which roles.

If a selectable function has a mandatory role mapping, this means that if the function is implemented it must be satisfied in a certain way; it does not mean that it is mandatory to implement that function.

If a selectable function does not have a mandatory role but is mapped to multiple optional roles, then at least one of them must be selected if the function is implemented.

Evaluation Activities

FMT_MOF.1
TSS
The evaluator shall examine the TSS to ensure it identifies the restrictions consistent with this requirement. For every claimed management function across all interfaces, the TSS must specify how the restriction is achieved and by whom.
Guidance
If the role restriction mechanism is configurable, the evaluator shall examine the operational guidance to determine that the necessary instructions to meet the requirement for the TOE in its evaluated configuration are provided. This applies only to management functions implemented by or accessible through the TSF.
Tests
Testing only applies to functions implemented by or accessible through the TSF. The evaluator shall, for each management function, assume the role defined for that function and demonstrate that the assigned role can perform the functions. The evaluator shall, for each management function, assume each role not assigned to that function, attempt to use the function, and verify that the TSF does not permit it. It may be necessary to perform multiple iterations of this test if the TOE has multiple interfaces that can be used to perform management functions.

FMT_SMF.1/STIP Specification of Management Functions

FMT_SMF.1.1/STIP

The TSF shall be capable of performing the following management functions **in support of STIP operations**: [

- *Ability to manage user accounts*
- *Ability to manage remote audit mechanism*
- *Ability to perform on-demand integrity tests*
- *Ability to import and remove X.509v3 certificates into/from the Trust Anchor Database*
- *Ability to configure identifying information for the TOE's embedded CA*
- *Ability to configure a maximum certificate validity duration*
- *Ability to manage inspection policy*
- *Ability to configure inspection processing [assignment: details beyond those covered by "inspection policy"]*

*[**selection**:*

- *Ability to configure local audit behavior,*
- *Ability to configure and manage certificate profiles,*
- *Ability to revoke issued certificates,*
- *Ability to configure certificate status services,*
- *Ability to configure automated process used to approve the revocation of a certificate or information about the revocation of a certificate,*
- *Ability to clear a cache of valid issued certificates,*
- *Ability to configure rules for automated issuance of certificates,*
- *Ability to modify the CRL configuration,*
- *Ability to modify the OCSP configuration,*
- *Ability to import private keys,*
- *Ability to configure the TOE's behavior on validating certificates whose revocation status cannot be determined,*
- *Ability to configure the TOE's behavior when non-supported critical extensions occur in a requested server certificate,*
- *Ability to generate and export PKCS#10 messages,*
- *Ability to generate and export EST messages and accept and process EST responses,*
- *Ability to configure TLS error responses for monitored clients,*
- *Ability to configure notification and consent message for monitored clients,*
- *Ability to configure rules for displaying a notification and consent message for acknowledgement prior to TLS inspection processing,*
- *Ability to search the certificate repository,*
- *No other capabilities*

]].

Application Note: This SFR defines mandatory and selectable management functions for STIP functionality specifically. The claims made here are in addition to those functions claimed for the Base-PP iteration of FMT_SMF.1.

Evaluation Activities

FMT_SMF.1/STIP
TSS
The evaluator shall examine the TSS to verify that it identifies the management functions that the TSF supports. If the TOE has multiple management interfaces, the evaluator shall verify that

the TSS identifies the management functions that are available on each interface.
Guidance
For each management function that can be performed on each management interface, the evaluator shall ensure that the operational guidance includes instructions on how an authorized administrator may perform the function, including any restrictions or limitations on the use of the function.
Tests
For each management function that can be performed on each management interface, the evaluator shall ensure that the operational guidance is sufficiently detailed to instruct an authorized administrator on how to perform that function.

FMT_SMR.2/STIP Restrictions on Security Roles

FMT_SMR.2.1/STIP

The TSF shall maintain the roles: [

- *Security Administrator,*

*[**selection**:*

- *Auditor,*
- *CA Operations Staff,*
- *Account Manager,*
- *no other roles*

]].

Application Note: This SFR is an iterated version of the FMT_SMR.2 SFR for the Base-PP. The purpose of this iteration is that the STIP functionality may be distributed across multiple administrative roles. If the TOE does not enforce role separation, the ST author selects "no other roles" to indicate that STIP functionality is managed by the same Security Administrator role specified in the Base-PP.

As is the case in the Base-PP, the TOE does not need its roles to have the same names as those defined in this SFR. It is expected that the ST will define the administrative roles and privileges defined by the TSF and map them to the roles listed in this PP-Module.

If “ability to configure local audit storage behavior” is selected in FMT_SMF.1/STIP, the ‘Auditor’ role must be selected here; role separation is required for audit storage functionality.

FMT_SMR.2.2/STIP

The TSF shall be able to associate users with roles.

FMT_SMR.2.3/STIP

The TSF shall ensure that the conditions [

- *All roles shall be able to administer the TOE locally,*
- *All roles shall be able to administer the TOE remotely,*

*[**selection**:*

- *No identity is authorized to assume both an Account Manager role and any of the other roles in FMT_SMR.2.1/STIP,,*
- *No identity is authorized to assume both an Auditor role and any of the other roles in FMT_SMR.2.1/STIP,,*
- *No other conditions*

]] are satisifed.

Application Note: This PP-Module refines the SFR defined in the Base-PP to include additional administrative roles. As defined in FMT_MOF.1, the TSF is expected to provide different privileges to the given roles.

If the TSF supports an Auditor and/or Account Manager role, it is expected that the relevant selections above will be made. It is the intent of this PP-Module that if either or both of these roles are provided, their critical functionality is isolated from any other roles (see FMT_MOF.1).

Evaluation Activities

FMT_SMR.2/STIP
TSS
The evaluator shall examine the TSS to verify that it identifies the management roles that the TOE maintains as well as any restrictions or limitations on the assignment of these roles (e.g. whether multiple roles can be assumed by the same user or if certain roles are mutually exclusive).

If the TOE supports multiple management roles, the evaluator shall verify that any differences in role enforcement between interfaces are discussed. For example, a TOE may have a local console that uses a separate administrative account from a remote GUI, such that any user who is authorized to use the local console is a Security Administrator, while the remote GUI maintains its own separate role structure.
Guidance
The evaluator shall examine the operational guidance to verify that it includes guidance on how to associate users with management roles.
Tests
For each supported management interface, the evaluator shall follow the operational guidance to associate user accounts with the management roles that are supported on those interfaces. If there are any restrictions on the assignment of management roles, such as the inability to assign two mutually exclusive roles to the same user, the evaluator shall attempt to violate these restrictions to verify that they are enforced.

Testing of the actual functional limitations of the assigned management roles is addressed by FMT_MOF.1.

5.2.7 Protection of the TSF (FPT)

FPT_FLS.1 Failure with Preservation of Secure State

FPT_FLS.1.1

The TSF shall preserve a secure state when the following types of failures occur: **DRBG failure, integrity test failure, external audit server is unavailable, [selection: *local audit storage is full, update signature verification failure, integrity failure on local audit, integrity failure on Trust Anchor database, [assignment: other potential TSF failures]*]**.

Application Note: The intent of this requirement is to prevent the use of failed randomization and other events that can compromise the operation of the TOE. This means that the TOE must be able to attain a secure/safe state when any of the identified failures occurs. If the TOE should encounter a failure in the middle of a critical operation, the TOE should not just quit operating, leaving key material and user data unprotected.

The failure of an Operational Environment component can be just as detrimental to security as a failure of the TSF itself. Therefore, in addition to describing the potential TSF failures and how the TOE preserves a secure state in response, the ST author is also expected to use this SFR to express how the TOE is made aware of any environmental failures and how it responds to these

Evaluation Activities

FPT_FLS.1
TSS
The evaluator shall examine the TSS to determine that the TOE's implementation of the fail secure functionality is documented, including all secure states for the TOE. The evaluator shall first examine the TSS section to ensure that all failure modes specified in the ST are described. The evaluator shall then ensure that the TOE will attain a secure state after inserting each specified failure mode type. The evaluator shall review the TSS to determine that the definition of secure state is defined and is suitable to ensure protection of key material and user data.
Guidance
The evaluator shall examine the operational guidance to ensure it describes the actions that might occur in response to any detected failures and provides remedial instructions for the administrator.
Tests
The evaluator shall attempt to cause each documented failure to occur and shall verify that the actions taken by the TSF are those specified in FPT_FLS.1.1. For those failures that the evaluator cannot cause, the evaluator shall provide a justification to explain why the failure could not be induced.

FPT_KST_EXT.1 No Plaintext Key Export

FPT_KST_EXT.1.1

The TSF shall prevent the plaintext export of [**assignment**: *list of all keys used by the TSF*].

Application Note: Keys include all TOE secret and private keys which includes keys generated for issued certificates. The intent of this requirement is to prevent the keys from being exported, even by a security administrator.

Evaluation Activities

FPT_KST_EXT.1
TSS
The evaluator shall examine the TSS to ensure it lists all keys and specifies what interfaces exist to export key data, if any.
Guidance
There are no guidance EAs for this component.
Tests
The evaluator shall access each export interface of the TOE, if any, and shall verify that the interface prevents the export of all keys listed in the TSS.

FPT_KST_EXT.2 TSF Key Protection

FPT_KST_EXT.2.1

The TSF shall prevent unauthorized use of all TSF private and secret keys.

Application Note: The intent of this requirement is to protect TSF private and secret keys from both unauthorized users, privileged users, and unprivileged processes. Keys specific to the TOE that should be addressed in this requirement include, but are not limited to, the TOE's embedded CA's private signing key, private keys associated to certificates issued by the TOE's embedded CA and TLS session keys established to facilitate inspection of traffic. Users should not be able to access the keys through "normal" interfaces. Processes that use private or secret keys to meet the functionality described in this PP module are considered authorized; all other processes are unauthorized. When an interface allows both authorized and unauthorized access to a key (for example, certificate signing functions with access to the embedded CA's private signing key are authorized only when certificate to be signed corresponds to a valid certificate belonging to a requested, and is unauthorized at any other time), evidence of protection includes logging of accesses via the common interface, as indicated in Table 2 for FAU_GEN.1.

Evaluation Activities

FPT_KST_EXT.2
TSS
The evaluator shall examine the TSS to ensure it describes how unauthorized use of TSF private and secret keys is prevented for both users and processes.
Guidance
The evaluator shall examine the AGD guidance to ensure it contains instructions for configuring the TOE or Operational Environment to prevent unauthorized access to TSF secret and private keys by users or processes.
Tests
The evaluator shall assume each of the non-Administrator roles supported by the TOE and shall attempt to use the available TOE interface to access the keys. The evaluator shall verify that these attempts fail.

FPT_RCV.1 Manual Trusted Recovery

FPT_RCV.1.1

After [**assignment**: *list of failures/service discontinuities*] the TSF shall enter a maintenance mode where the ability to return to a secure state is provided.

Application Note: This requirement ensures that the TSF can determine that the TOE is started up without protection compromise and can recover without protection compromise after discontinuity of operations. Anticipated failures include actions that result in a system crash, media failures, or discontinuity of operations caused by erroneous administrative action or lack of erroneous administrative action. The data that needs to be restored includes the TSF keys needed for signature, the Trust Anchor Database, keys needed for management of certificates, all signed certificates, and any certificate status information

Evaluation Activities

FPT_RCV.1
TSS
The evaluator shall examine the TSS to determine that, for each failure or service discontinuity identified in the SFR, it describes how the TOE enters a maintenance mode after a failure and the possible actions that can take place while in that mode.

Guidance
The evaluator shall examine the AGD guidance to ensure it contains instructions for restoring the TOE to a secure state when it enters the maintenance mode, including the steps necessary to perform while in this state.
Tests
The evaluator shall attempt to cause each documented failure to occur and shall verify that the result of this failure is that the TSF enters a maintenance mode. The evaluator shall also verify that the maintenance mode can be exited and the TSF can be restored to a secure state. This testing may be performed in conjunction with FPT_FLS.1.

5.3 TOE Security Functional Requirements Rationale

The following rationale provides justification for each security objective for the TOE, showing that the SFRs are suitable to meet and achieve the security objectives:

Table 4: SFR Rationale

Objective	Addressed by	Rationale
O.AUDIT_LOSS_RESPONSE	FAU_STG.4	This SFR supports the objective by requiring the TSF to disable the execution of auditable events if the audit trail cannot be written to.
O.AUDIT_PROTECTION	FAU_STG.1 (from Base-PP)	This SFR supports the objective by ensuring that stored audit records are protected against unauthorized modification or destruction.
	FAU_SAR.1 (optional)	This SFR supports the objective by ensuring that stored audit records are protected against unauthorized access.
O.CERTIFICATES	FIA_X509_EXT.1/Rev (from Base-PP)	This SFR supports the objective by defining the TOE functionality for certificate validation.
	FIA_X509_EXT.3 (from Base-PP)	This SFR supports the objective by defining the mechanism by which the TOE generates certificate signing requests, which includes validation of the certificate provided in response.
	FDP_CER_EXT.1	This SFR supports the objective by defining the rules the TOE must use to generate and issue proxy TLS server certificates from its internal CA.
	FDP_CER_EXT.2	This SFR supports the objective by requiring the TOE to link the certificates presented for TLS connectivity with the certificates it issues from its internal CA.
	FDP_CER_EXT.3	This SFR supports the objective by defining the rules for the TOE's issuing of proxy TLS server certificates.
	FDP_CSIR_EXT.1	This SFR supports the objective by defining how the TOE can ensure the use of fresh certificates.
	FIA_ENR_EXT.1	This SFR supports the objective by defining the mechanism by which the TOE requests a certificate for its own embedded CA's signing key.
	FIA_X509_EXT.1/STIP	This SFR supports the objective by defining the certificate validation rules that must be followed for certificates that are used for proxy TLS connections.
	FIA_X509_EXT.2/STIP	This SFR supports the objective by

		defining the certificate authentication behavior for STIP connections.
	FDP_PIN_EXT.1 (optional)	This SFR supports the objective by defining the optional implementation of certificate pinning.
	FIA_ESTC_EXT.2 (optional)	This SFR supports the objective by defining requirements for the composition of EST requests if the TOE supports EST.
	FDP_CER_EXT.4 (selection-based)	This SFR supports the objective by defining the rules the TOE must use to generate and issue proxy TLS client certificates from its internal CA if mutual authentication is supported.
	FDP_CER_EXT.5 (selection-based)	This SFR supports the objective by defining the rules for the TOE's issuing of proxy TLS client certificates if mutual authentication is supported.
	FDP_CRL_EXT.1 (selection-based)	This SFR supports the objective by defining rules for the generation of CRLs if the TOE uses this as the mechanism to ensure the freshness of its issued certificates.
	FDP_CSI_EXT.1 (selection-based)	This SFR supports the objective by defining the revocation echecking method supported by the TOE for the proxy TLS server certificates it issues, if revocation is how the freshness of its issued certificates is assured.
	FDP_CSI_EXT.2 (selection-based)	This SFR supports the objective by defining the revocation echecking method supported by the TOE for the proxy TLS client certificates it issues, if mutual authentication is supported and revocation is how the freshness of its issued certificates is assured.
	FDP_OCSP_EXT.1 (selection-based)	This SFR supports the objective by defining rules for the generation of OCSP responses if the TOE uses this as the mechanism to ensure the freshness of its issued certificates.
	FDP_OCSPS_EXT.1 (selection-based)	This SFR supports the objective by defining rules for the implementation of OCSP stapling if the TOE supports this functionality.
	FIA_ESTC_EXT.1 (selection-based)	This SFR supports the objective by defining requirements for the implementation of EST if the TOE uses this mechanism to obtain TLS certificates for its own use.
O.DISPLAY_BANNER	FTA_TAB.1 (from Base-PP	This SFR supports the objective by applying a warning banner to any interface used by an administrator to access the TOE.
	FTA_TAB.1/TLS (selection-based)	This SFR supports the objective by optionally applying a warning banner to a user whose network activity passes through the TOE for decryption and potential inspection.
O.PERSISTENT_KEY_PROTECTION	FCS_STG_EXT.1	This SFR supports the objective by requiring the TOE to implement hardware-based protection for stored

		keys.
	FDP_STG_EXT.1	This SFR supports the objective by defining the mechanism used to protect public key data from unauthorized modification.
	FPT_KST_EXT.1	This SFR supports the objective by requiring the TSF to enforce the prevention of plaintext key export.
	FPT_KST_EXT.2	This SFR supports the objective by preventing the unauthorized use of secret and private keys.
	FCS_CKM_EXT.5 (selection-based)	This SFR supports the objective by defining the integrity mechanism used to guarantee the integrity of public key data.
O.PROTECTED_COMMUNICATIONS	FCS_CKM.4 (from Base-PP)	This SFR supports the objective by ensuring secret and private key data is disposed of immediately after use to prevent unauthorized disclosure of keys.
	FCS_TLSC_EXT.1 (from Base-PP)	This SFR supports the objective by defining requirements for the TOE's implementation of TLS as a client.
	FCS_TLSS_EXT.1 (from Base-PP)	This SFR supports the objective by defining requirements for the TOE's implementation of TLS as a server.
	FTP_ITC.1 (refined from Base-PP)	This SFR supports the objective by defining the TOE interfaces that require protected communications as well as the methods of protection applied to these interfaces.
	FCS_COP.1/STIP	This SFR supports the objective by defining cryptographic algorithms the TOE must support for decryption and re-encryption of proxy TLS traffic.
	FCS_TTTC_EXT.1	This SFR supports the objective by defining requirements for the TOE's implementation of TLS as a client, specifically in the case where the TOE is establishing a proxy connection between itself and the original requested TLS server.
	FCS_TTTC_EXT.5	This SFR supports the objective by defining the Supported Groups used by the TOE's proxy TLS client interface.
	FCS_TTTS_EXT.1	This SFR supports the objective by defining requirements for the TOE's implementation of TLS as a server, specifically in the case where the TOE is establishing a proxy connection between itself and the original monitored TLS client.
	FDP_PPP_EXT.1	This SFR supports the objective by defining the processing rules that the TOE applies to plaintext traffic once decrypted.
	FDP_PRC_EXT.1	This SFR supports the objective by defining requirements for the routing of decrypted plaintext traffic.
	FDP_STIP_EXT.1	This SFR supports the objective by defining the TOE's ability to establish proxy TLS sessions between a

		monitored client and a requested server and to apply appropriate rules to the handling of the decrypted traffic.
	FDP_TEP_EXT.1	This SFR supports the objective by defining the TOE's abbility to enforce filtering rules on TLS traffic passing through the TOE.
	FCS_TTTC_EXT.3 (selection-based)	This SFR supports the objective by defining optional support for TLS mutual authentication that is applied to the TOE's proxy TLS client interface.
	FCS_TTTC_EXT.4 (selection-based)	This SFR supports the objective by defining optional support for TLS session renegotiation that is applied to the TOE's proxy TLS client interface.
	FCS_TTTS_EXT.3 (selection-based)	This SFR supports the objective by defining optional support for TLS mutual authentication that is applied to the TOE's proxy TLS server interface.
	FCS_TTTS_EXT.4 (selection-based)	This SFR supports the objective by defining optional support for TLS session renegotiation that is applied to the TOE's proxy TLS server interface.
	FDP_STIP_EXT.2 (selection-based)	This SFR supports the objective by defining the optional capability of the TOE to establish a proxy TLS session in the case where mutual authentication is supported.
O.RECOVERY	FPT_FLS.1	This SFR supports the objective by requiring the TSF to preserve a secure state when certain failures occur.
	FPT_RCV.1	This SFR supports the objective by requiring the TSF to support a maintenance mode of operation that is entered when certain failures occur.
O.RESIDUAL_INFORMATION_CLEARING	FDP_RIP.1	This SFR supports the objective by defining the residual data that is cleared from TOE memory and when the clearing occurs.
O.SYSTEM_MONITORING	FAU_STG_EXT.1 (from Base-PP)	This SFR supports the objective by defining a mechanism for the secure storage of audit data in the OE.
	FAU_GEN.1/STIP	This SFR supports the objective by defining the auditable events specific to STIP functionality that the TSF must generate.
	FAU_GCR_EXT.1	This SFR supports the objective by defining the mechanism the TOE uses to store certificate data.
	FAU_SAR.3 (optional)	This SFR supports the objective by optionally defining the functionality to search audit records for events associated with a particular certificate.
	FAU_SCR_EXT.1 (selection-based)	This SFR supports the objective by requiring the TOE to implement a search function for certificate storage if the TSF implements its own certificate store (as opposed to relying on environmental storage).
O.TOE_ADMINISTRATION	FMT_MOF.1	This SFR supports the objective by defining the authorized use of the TOE

	by association between the supported management functions and the roles that are authorized to perform them.
FMT_SMF.1/STIP	This SFR supports the objective by defining the TOE's management functions that are specific to STIP functionality.
FDP_SMR.2/STIP	This SFR supports the objective by defining additional management roles that the TOE may support that are specific to STIP functionality.

5.4 TOE Security Assurance Requirements

This PP-Module does not define any SARs beyond those defined within the Base-PPs to which it can claim conformance. It is important to note that a TOE that is evaluated against this PP-Module is inherently evaluated against the General Purpose Operating Systems PP as well. This PP includes a number of EAs associated with both Security Functional Requirements (SFRs) and SARs. Additionally, this PP-Module includes a number of SFR-based EAs that similarly refine the SARs of the Base-PPs. The evaluation laboratory will evaluate the TOE against the chosen Base-PP and supplement that evaluation with the necessary SFRs that are taken from this PP-Module.

6 Consistency Rationale

6.1 Protection Profile for General Purpose Operating Systems

6.1.1 Consistency of TOE Type

If this PP-Module is used to extend the NDcPP, the TOE type for the overall TOE is still a generic network device. However, one of the functions of the device must be the ability for it to act as an SSL/TLS Inspection Proxy. The TOE boundary is simply extended to include that functionality.

6.1.2 Consistency of Security Problem Definition

The threats, assumptions, and OSPs defined by this PP-Module (see sections 3.1 through 3.3) supplement those defined in the NDcPP as follows:

PP-Module Threat, Assumption, OSP	Consistency Rationale
T.UNTRUSTED_COMMUNICATION	The threat of untrusted communication can provide unauthorized access to unintended resources if using weak cryptography or use untrusted intermediate systems. This can be mitigated either by protocols defined in this PP-Module or in the Base-PP.
T.AUDIT	Auditing poses a threat if certain activities aren't logged, like the issuance of certificates. This threat can be mitigated if proper configurations are in place to prevent the compromise of audit data defined in this PP-Module or the Base-PP.
T.UNAUTHORIZED_USERS	The threat of unauthorized users attempting to gain access to other users' credentials can be addressed by placing protections for logged-in users and only allow privileged user access methods defined in this PP-Module or in the Base-PP.
T.CREDENTIALS	Beyond the Base-PP, the threat of manipulation of the CA signing key can be mitigated by providing access protection to persistent keys.
T.SERVICES	The threat of misuse or manipulation of services is not defined in the Base-PP, but it is consistent with the general threat of unauthorized manipulation of the TSF.
T.DEVICE_FAILURE	The failure of the certificate authority or routing traffic to inspection poses a threat not defined in the Base-PP.
T.UNAUTHORIZED_DISCLOSURE	The Base-PP does not include the threat of unauthorized disclosure to sensitive data that is only intended for the monitored client because this is an interface that the Base-PP cannot assume all conformant TOEs have.
T.INAPPROPRIATE_ACCESS	The threat of inappropriate access to unintended servers could disclose unauthorized traffic to inspection processes which is not defined in the Base-PP because a generic network device does not necessarily have a traffic inspection functionality.
P.AUTHORIZATION_TO_INSPECT	The Base-PP cannot define the interactions that an end user will have with a generic device because it may vary depending on the specific device type. This PP-Module defines a policy that is specific to the use case of a STIP device.

6.1.3 Consistency of Objectives

The objectives for the TOEs are consistent with the General Purpose Operating Systems PP based on the following rationale:

PP-Module TOE Objective	Consistency Rationale
O.AUDIT_LOSS_RESPONSE	The Base-PP does not define any TOE objectives so PP-Module objectives do not conflict with it.
O.AUDIT_PROTECTION	The Base-PP does not define any TOE objectives so PP-Module objectives do not conflict with it.
O.CERTIFICATES	The Base-PP does not define any TOE objectives so PP-Module objectives do not conflict with it.
O.DISPLAY_BANNER	The Base-PP does not define any TOE objectives so PP-Module

	objectives do not conflict with it.
O.PERSISTENT_KEY_PROTECTION	The Base-PP does not define any TOE objectives so PP-Module objectives do not conflict with it.
O.PROTECTED_COMMUNICATIONS	The Base-PP does not define any TOE objectives so PP-Module objectives do not conflict with it.
O.RECOVERY	The Base-PP does not define any TOE objectives so PP-Module objectives do not conflict with it.
O.RESIDUAL_INFORMATION_CLEARING	The Base-PP does not define any TOE objectives so PP-Module objectives do not conflict with it.
O.SYSTEM_MONITORING	The Base-PP does not define any TOE objectives so PP-Module objectives do not conflict with it.
O.TOE_ADMINISTRATION	The Base-PP does not define any TOE objectives so PP-Module objectives do not conflict with it.

The objectives for the TOE's Operational Environment are consistent with the General Purpose Operating Systems PP based on the following rationale:

PP-Module Operational Environment Objective	Consistency Rationale
OE.AUDIT	This objective intends for the TOE’s OE to have adequate storage to retain the TOE's audit records. This objective is not defined in the Base-PP but can be assumed to be consistent with the Base-PP because FAU_STG_EXT.1 requires transmission of audit data to an environmental audit server, which means that there should be some assurance of the security of that server.
OE.CERT_REPOSITORY	This objective intends for the TOE’s OE to provide a certificate repository. This is not defined in the Base-PP because not all network devices will necessarily need to interface with a certificate repository.
OE.CERT_REPOSITORY_SEARCH	This objective intends for the TOE’s OE which will provide a certificate repository to also have the capability to search within the repository. This is not defined in the Base-PP because not all network devices will necessarily need to interface with a certificate repository.

6.1.4 Consistency of Requirements

This PP-Module identifies several SFRs from the General Purpose Operating Systems PP that are needed to support SSL/TLS Inspection Proxies functionality. This is considered to be consistent because the functionality provided by the General Purpose Operating Systems PP is being used for its intended purpose. The PP-Module also identifies a number of modified SFRs from the General Purpose Operating Systems PP that are used entirely to provide functionality for SSL/TLS Inspection Proxies. The rationale for why this does not conflict with the claims defined by the General Purpose Operating Systems PP are as follows:

PP-Module Requirement	Consistency Rationale
	Modified SFRs
FAU_STG.1	Other than this SFR becoming mandatory versus optional, there is no modification to this SFR.
FCS_CKM.4	The ST author is instructed to include security critical parameters and when key destruction is required.
FCS_TLSC_EXT.1	Other than this SFR becoming mandatory versus selection-based, there is no modification to this SFR.
FCS_TLSS_EXT.1	Other than this SFR becoming mandatory versus selection-based, there is no modification to this SFR.
FIA_X509_EXT.1/Rev	Other than this SFR becoming mandatory versus selection-based, there is no modification to this SFR.
FIA_X509_EXT.2	The PP-Module partially completes selections in this SFR using the available options to specify minimum required functionality for X.509 authentication based on its use in STIP. The PP-Module also refines the authorized management roles that can perform the function defined in FIA_X509_EXT.2.2.

FIA_X509_EXT.3	There is no change to this SFR. Only its trigger for inclusion is changed because this PP-Module introduces an alternate method of obtaining a certificate for the TOE.
FTP_ITC.1	The PP-Module partially completes selections and assignments in this SFR using the available options to specify external interfaces and trusted channels that all STIP products must support at minimum.

Additional SFRs

This PP-Module does not add any requirements when the General Purpose Operating Systems PP is the base.

Mandatory SFRs

FAU_GCR_EXT.1	This SFR applies to storing certificates in a certificate repository which is not listed in the Base-PP.
FAU_GEN.1/STIP	This SFR adds new auditable events for the TOE that relate to the functionality that is introduced by the PP-Module
FAU_STG.4	This SFR applies to the prevention of audit data loss by the inclusion of the auditor role which is not listed in the Base-PP.
FCS_COP.1/STIP	This SFR provides encryption/decryption cipher suites used in support for the through-traffic processing of the TOE.
FCS_STG_EXT.1	This SFR applies to the storage of persistent private and secret keys which is not defined in the Base-PP.
FCS_TTTC_EXT.1	This SFR applies to thru-traffic TLS inspection client protocol which is not defined in the Base-PP.
FCS_TTTC_EXT.5	This SFR applies to client supported groups extension for thru-traffic TLS inspection.
FCS_TTTS_EXT.1	This SFR applies to thru-traffic TLS inspection server protocol which is not defined in the Base-PP.
FDP_CER_EXT.1	This SFR applies to the implementation of certificate profile functionality for server certificates which is not defined in the Base-PP.
FDP_CER_EXT.2	This SFR applies to the establishing and recording a linkage from validated to issued certificates which is not defined in the Base-PP.
FDP_CER_EXT.3	This SFR applies to rules for the issuance of certificates which is not defined in the Base-PP.
FDP_CSIR_EXT.1	This SFR applies to the ability to generate certificate status information if the validity period can be configured to last longer than 24 hours.
FDP_PPP_EXT.1	This SFR applies to the enforcement of the TLS processing policy which is not defined in the Base-PP.
FDP_PRC_EXT.1	This SFR applies to the routing of information flows containing plaintext which is not defined in the Base-PP.
FDP_RIP.1	This SFR applies to providing the capability to allocation or deallocation of resources which in this PP-Module is any data buffers used to implement STIP functionality which is not defined in the Base-PP.
FDP_STG_EXT.1	This SFR enforces protection of trusted public keys and certificates implemented using access control or integrity mechanism which is not defined in the Base-PP.
FDP_STIP_EXT.1	This SFR applies to STIP-specific processing operations which are not defined in an RFC or specified in the Base-PP.
FDP_TEP_EXT.1	This SFR applies to the enforcement of the TLS session establishment policy which is not defined by the Base-PP.
FIA_ENR_EXT.1	This SFR applies to the ability to generate a certificate request which is not defined in the Base-PP.
FIA_X509_EXT.1/STIP	This SFR specifies validation of certificates used for connections supporting STIP functions.
FMT_MOF.1	This SFR applies to the restriction of management functions to certain roles which are not defined in the Base-PP which only requires management functionality to be

	performed by a security administrator.
FMT_SMF.1/STIP	This SFR defines additional management functions to address the STIP functionality not defined in the Base-PP.
FMT_SMR.2/STIP	This SFR defines additional management roles that the TOE may define to enforce role separation, which is a security enhancement on the Base-PP's requirement that only one management role is necessary.
FPT_FLS.1	This SFR applies to preserving a secure state when different failures occur which is not defined in the Base-PP.
FPT_KST_EXT.1	This SFR applies to the prevention of plaintext key export which is not defined in the Base-PP.
FPT_KST_EXT.2	This SFR applies to the prevention of unauthorized use of private and secret keys which is not defined in the Base-PP.
FPT_RCV.1	This SFR applies to the maintenance mode that provides the ability to return to a secure state is provided which is not defined in the Base-PP.
Optional SFRs	
FAU_SAR.1	This SFR applies to who can view all the audit records which includes the added role of the auditor, which is not defined in the Base-PP.
FAU_SAR.3	This SFR applies to the ability to search within audit records based on various identifiers which is not defined in the Base-PP.
FDP_PIN_EXT.1	This SFR applies to certificate pinning which is not defined in the Base-PP.
Selection-based SFRs	
FDP_CRL_EXT.1	This SFR applies to the revocation of certificates which is not defined in the Base-PP.
FDP_CSI_EXT.1	This SFR applies to generating certificate status information which is not defined in the Base-PP.
FDP_OCSP_EXT.1	This SFR applies to generating OCSP responses which is not defined in the Base-PP.
FCS_OCSPS_EXT.1	This SFR applies to OCSP stapling which is not defined in the Base-PP.
FIA_ESTC_EXT.1	This SFR applies to the enforcement of Enrollment of Secure Transport to obtain its embedded CA certificate which is not defined in the Base-PP.
FTA_TAB.1/TLS	This SFR applies to having a notice and consent warning message at the start of an SSL/TLS inspection session which is not defined in the Base-PP.
FCS_TTTC_EXT.3	This SFR applies to thru-traffic TLS Inspection Client Protocol with mutual authentication which is not defined in the Base-PP.
FCS_TTTS_EXT.3	This SFR applies to thru-traffic TLS Inspection Server Protocol with mutual authentication which is not defined in the Base-PP.
FDP_CER_EXT.4	This SFR applies to the implementation of the certificate profile functionality for TLS client certificates.
FDP_CER_EXT.5	This SFR applies to the certificate issuance rules applied for client certificates which is not defined in the Base-PP.
FDP_CSI_EXT.2	This SFR applies to generating certificate status information for issued client certificates which is not defined in the Base-PP.
FDP_STIP_EXT.2	This SFR applies to the TLS session implementation of the inspection operation that is not defined in the Base-PP.
FAU_SCR_EXT.1	This SFR applies to providing the capability to search the certificate repository which is not defined by the Base-PP.
FCS_CKM_EXT.5	This SFR applies to the protection of persistent public keys from undetected modification which is not defined in the Base-PP.
FCS_TTTC_EXT.4	This SFR applies to session renegotiation for thru-traffic TLS inspection (client-side).
FCS_TTTS_EXT.4	This SFR applies to session renegotiation for thru-traffic TLS inspection (server-side).

Objective SFRs	
FIA_ESTC_EXT.2	This SFR applies to the generation of TLS unique values used by client which is not defined in the Base-PP.

Implementation-Dependent SFRs
This PP-Module does not define any Implementation-Dependent requirements.

6.2 TOE Security Assurance Requirements

This PP-Module does not define any Security Assurance requirements. The SARs from the Base-PP must be satisfied.

Appendix A - Optional SFRs

A.1 Strictly Optional Requirements

A.1.1 Persistent Local Audit Storage

The SFRs in this section are optional. They should be claimed if the TOE provides local audit storage (i.e., if FAU_STG.1 is claimed in the base NDcPP) and that local audit storage is intended to provide a persistent, searchable record of security events within the TOE, either as a backup or replacement of an external audit capability.

FAU_SAR.1 Audit Review

FAU_SAR.1.1

The TSF shall provide [**selection**: *Security Administrators*, *Auditors*] with the capability to read all information from the local audit records

FAU_SAR.1.2

The TSF shall provide the local audit records in a manner suitable for the administrator to interpret the information.

Evaluation Activities

FAU_SAR.1
This activity should be accomplished in conjunction with the testing of FAU_GEN.1. Review of each of the generated audit records demonstrates that these records are reviewable.

FAU_SAR.3 Selectable Audit Review

FAU_SAR.3.1

The TSF shall provide the ability to apply searches of local audit data based on **[assignment: *object identifier of certificate*] associated with the event**.

Application Note:

Evaluation Activities

FAU_SAR.3
This activity should be accomplished in conjunction with the testing of FAU_GEN.1.

A.1.2 Certificate Pinning

Certificate pinning is an optional feature to address the threat of unauthorized access to user data managed by the TOE via unauthorized STIP or adversary man-in-the-middle exploits. This feature is desirable since implementation of a STIP to protect a client enclave will prevent the clients from effectively providing this feature.

FDP_PIN_EXT.1 Certificate Pinning

FDP_PIN_EXT.1.1

The TSF shall be able to detect and [**selection**: *alert*, *[**assignment**: perform a [Security Administrator] managed action]*] to changes in the [**selection**: *public key*, *certificate*, *certificate issuer*] used by requested servers according to **[selection: *a Security Administrator configurable number of the most common requested servers, a Security Administrator specified list of servers, [assignment: a Security Administrator configurable rules based on attributes of the certificates used by the server]*]** .

Application Note: This requirement should be claimed if implemented by the TOE. If claimed, additional FMT_MOF.1 and audit events associated with the function must be claimed.

Evaluation Activities

FDP_PIN_EXT.1
TSS
The evaluator shall review the TSS to ensure the certificate pinning function is described.
Guidance
The evaluator shall review the AGD guidance to ensure it contains instructions for any

configurable aspects of the certificate pinning function.
Tests
The evaluator shall establish a monitored client and requested server with multiple certificates issued by one or more external certification authorities. The evaluator shall configure the TSF, to either pin one of the certificates, or to pin on the first certificates seen, and to alert on differences between the issued certificates for the requested server. If caching is supported, the evaluator shall either disable caching, or clear cache between TLS requests from the client. The evaluator shall then use the client to request a TLS session with the server using the first of the certificates, and observe that the pinning response is not observed. The evaluator shall then configure the server to use the second certificate, make a second request from the monitored client, and observe that the pinning alert response is observed.

A.2 Objective Requirements

A.2.1 Identification and Authentication (FIA)

FIA_ESTC_EXT.2 Client Use of TLS-Unique Value

FIA_ESTC_EXT.2.1

The TSF shall generate tls-unique values and integrate them into EST requests it generates in accordance with RFC 7030 section 3.5.

Application Note: This SFR describes an optional element of RFC 7030 that strengthens the authentication provided by EST. While RFC 7030 requires EST servers to validate the tls-unique values when presented, this requirement is not implemented in current EST servers. FIA_ESTC_EXT.2.1 will be integrated into FIA_ESTC_EXT.1 in a subsequent release of this PP-Module and should be claimed if the EST implementation supports it.

Evaluation Activities

FIA_ESTC_EXT.2
TSS
The evaluator shall examine the TSS to ensure the description of EST includes implementation of TLSunique values.
Guidance
The evaluator shall examine the operational guidance to ensure it contains instructions on configuring the TOE so that EST conforms to the description in the TSS, to include any configuration associated to the inclusion of TLS-unique values in certificate requests.
Tests
The evaluator shall follow guidance documentation to implement the EST request function to include TLS-unique values in the certificate request. The evaluator shall establish trust with an external EST server and associated CA and submit a simple certificate request. The evaluator shall review the request received by the EST server and observe that the request contains the TLS-unique value and that it matches the TLS-unique value established under the TLS session.

A.3 Implementation-Based Requirements

This PP-Module does not define any Implementation-Based SFRs.

Appendix B - Selection-Based Requirements

B.1 Certificate Status Information

FDP_CRL_EXT.1 Certificate Revocation List Generation

The inclusion of this selection-based component depends upon selection in FDP_CSI_EXT.1.1.

FDP_CRL_EXT.1.1

When the TSF is configured to generate CRLs, the TSF shall verify that all mandatory fields in any generated CRL contains values in accordance with ITU-T Recommendation X.509. At a minimum, the following items shall be validated:

a. If the version field is present, then it shall contain a 1.
b. If the CRL contains any critical extensions, then the version field shall be present and contain the integer 1.
c. The [**selection**: *issuer*, *issuerAltName*] fields shall indicate the configured name of the CA.
d. If the issuer field contains a null Name (e.g., a sequence of zero relative distinguished names), then the CRL shall contain a critical issuerAltName extension.
e. The signature and signatureAlgorithm fields shall contain the OID for a digital signature algorithm in accordance with FCS_COP.1**/SigGen in the NDcPP**.
f. The thisUpdate field shall indicate the issue date of the CRL.
g. The time specified in the nextUpdate field (if populated) shall not precede the time specified in the thisUpdate field.

Application Note: This requirement should be claimed if 'ITU-T Recommendation X.509v2 CRL' is selected in FDP_CSI_EXT.1.1

Evaluation Activities

FDP_CRL_EXT.1
TSS
The evaluator shall examine the TSS to ensure it indicates whether the TOE supports CRL generation and, if so, describes the CRL generation function. In addition, the evaluator shall ensure that the TSS identifies which of the values identified in FDP_CRL_EXT.1.1 can be included in CRLs.
Guidance
If the TOE supports configuration of the CRL issuing function, the evaluator shall examine the operational guidance to ensure that instructions are available to configure issuance of CRL in accordance with FDP_CRL_EXT.1.1.
Tests
The evaluator shall perform the following tests:

- ***Test 1:*** *The evaluator shall configure the CRL function using available user guidance and request a CRL in order to ensure that the resulting CRL satisfies all field constraints in FDP_CRL_EXT.1.1.*
- ***Test 2:*** *For each field defined in FDP_CRL_EXT.1.1, the evaluator shall attempt to create a CRL that violates the required conditions of the field. The evaluator shall determine that all such attempts are rejected by the TSF.*
- ***Test 3:*** *The evaluator shall make a selection of fields from a configured CRL function and shall attempt to create a CRL that violates the required conditions of the field. The evaluator shall determine that all such attempts are rejected by the TSF.*

FDP_CSI_EXT.1 Certificate Status Information

The inclusion of this selection-based component depends upon selection in FDP_CSIR_EXT.1.1.

FDP_CSI_EXT.1.1

The TSF shall generate certificate status information whose format complies with [**selection**: *ITU-T Recommendation X.509v2 CRL, the OCSP standard as defined by RFC 6960*].

FDP_CSI_EXT.1.2

The TSF shall support changes to the status of a certificate by [**selection**:

- *[**selection**: Security Administrator, CA Operations Staff]* ,
- *[**assignment**: automated revocation rules]*

].

FDP_CSI_EXT.1.3

The TSF shall [**selection**: *provide*, *interface with the Operational Environment to provide*] certificate status information generated in accordance with FDP_CSI_EXT.1.1 via [**selection**: *posting CRLs at the location specified in the cRLDistributionPoints of the issued certificate*, *an OCSP mechanism indicated in the authorityInfoAccess extension of the issued certificate*, *OCSP Stapling in accordance with FDP_OCSPS_EXT.1*].

Application Note: This SFR should be claimed if claimed if the selection 'generate certificate status information' is selected in FDP_CSIR_EXT.1.1.

The ST should specify the format(s) used to supply certificate status information in FDP_CSI_EXT.1.1, and the mechanism(s) used to provide the status to relying parties including all monitored clients in the second selection in FDP_CSI_EXT.1.3. If CRLs are identified in FDP_CSI_EXT.1.1, then cRLDistributionPoints must be claimed in FDP_CSI_EXT.1.3 and in FDP_CER_EXT.1.2 item (g), sub-item (e). If the OCSP standard is selected in FDP_CSI_EXT.1.1, then at least one of the last two options in the second selection of FDP_CSI_EXT.1.3 must be claimed. If the second option (OCSP) is claimed, authorityInfoAccess must be claimed in FDP_CER_EXT.1.2 item (g), sub-item (e). OCSP stapling may also be claimed if the TOE only generates CRLs, but interfaces with an external OCSP responder that uses those CRLs.

Automated rules for revoking certificates in response to the TOE's discovery that a previously issued certificate is no longer appropriate for the subject, or due to cache clearing, timeouts, or other rules should be described in the assignment of FDP_CSI_EXT.1.2.

Evaluation Activities

FDP_CSI_EXT.1
TSS
The evaluator shall examine the TSS to ensure it describes the certificate status function and applicable formats, in accordance with this requirement, that can be used to issue certificate status. The TSS must reflect the selection made by the ST author as well as the selection-based requirements from Appendix B.1 of the STIP PP-Module. The evaluator shall also ensure that the TSS describes the process for approving changes to the status of a certificate, including the interfaces that must be used.

If OCSP stapling is selected in FDP_CSI_EXT.1.3, but only CRLs are generated (OCSP responses are not generated by the TSF) as indicated in FDP_CSI_EXT.1.1, the evaluator shall examine the operational guidance to ensure it describes the interfaces to the operational environment required to generate the responses.
Guidance
The evaluator shall examine the operational guidance to ensure that it contains instructions for any configuration aspects of the certificate status change function and the steps needed to perform an approval, as well as any configuration required for interfaces to external certificate status providers.
Tests
Based on the selections, the evaluator shall perform the following tests. It is recommended that these be performed in conjunction with applicable tests associated with the requirements claimed in Appendix B.1 of the STIP PP-Module:

- ***Test 1:*** *For each certificate status format identified in FCS_CSI_EXT.1.1, the evaluator shall issue a valid certificate from the TOE. The evaluator shall then cause the TOE to issue certificate status information. The evaluator shall check the certificate status information using all indicated methods identified in FCS_CSI_EXT.1.3 to verify that each reflects that the certificate is valid.*
- ***Test 2:*** *For each selected certificate status format (CRLv2 or OCSP) identified in FCS_CSI_EXT.1.1, and for each mechanism indicated in FDP_CSI_EXT.1.2, the evaluator shall cause a valid certificate from the TOE to be revoked. The evaluator shall then cause the TOE to issue certificate status information. The evaluator shall check the certificate status information using all methods (cRLDistributionPoints, authorityInfoAccess, or OCSP Stapling) identified in FCS_CSI_EXT.1.3 to verify that each method reflects that the certificate is revoked.*

FDP_OCSP_EXT.1 OCSP Basic Response Generation

The inclusion of this selection-based component depends upon selection in FDP_CSI_EXT.1.1.

FDP_OCSP_EXT.1.1

When the TSF is configured to generate OCSP responses of the basic response type, the TSF shall ensure that all mandatory fields in the OCSP basic response contain values in accordance with RFC 6960. At a minimum, the following items shall be validated:

a. The version field shall contain a 0.
b. The signatureAlgorithm field shall contain the object identifier (OID) for a digital signature algorithm in accordance with FCS_COP.1**/SigGen in the NDcPP**.
c. The thisUpdate field shall indicate the time at which the status being indicated is known to be correct.
d. The producedAt field shall indicate the time at which the OCSP responder signed the response.
e. The time specified in the nextUpdate field (if populated) shall not precede the time specified in the thisUpdate field.

Application Note: This requirement should be claimed if 'the OCSP standard as defined by RFC 6960' is selected in FDP_CSI_EXT.1.1.

Evaluation Activities

FDP_OCSP_EXT.1
TSS
The evaluator shall examine the TSS to ensure it indicates whether the TOE supports OCSP and, if so, describes the OCSP response function. In addition, the evaluator shall ensure that the TSS identifies which of the values identified in FDP_OCSP_EXT.1.1 can be included in OCSP responses.
Guidance
If the TOE supports configuration of the OCSP function, the evaluator shall examine the operational guidance to ensure that instructions are available to configure the OCSP response function in accordance with FDP_OCSP_EXT.1.1.
Tests
The evaluator shall perform the following tests:

- ***Test 1:*** *The evaluator shall configure the OCSP response function, establish a monitored client and shall, in turn, cause an OCSP response by the TSF for the status of a certificate issued by the TOE's embedded CA which has not been revoked, a certificate issued by the TOE's embedded CA which has been revoked, and a certificate not issued by the TOE's embedded CA. The evaluator shall ensure that the response satisfies all constraints in FDP_OCSP_EXT.1.1 and provides an accurate status indication in accordance with RFC 6960.*
- ***Test 2:*** *For each of the constraints in FDP_OCSP_EXT.1.1, the evaluator shall attempt to create an OCSP response that violates the constraints. The evaluator shall determine that all such attempts are rejected by the TSF.*

FCS_OCSPS_EXT.1 OCSP Stapling

The inclusion of this selection-based component depends upon selection in FDP_CSI_EXT.1.3.

FCS_OCSPS_EXT.1.1

The TSF shall be able to process [**selection**: *Certificate Status Request extension in accordance with RFC 6066 section 8*, *Certificate Status Request List V2 in accordance with RFC 6961*].

FCS_OCSPS_EXT.1.2

The TSF shall [**selection**: *generate OCSP response information in accordance with FDP_OCSP_EXT.1*, *interface with an OCSP provider to obtain an OCSP response*] and populate a Certificate Status Message in accordance with RFC 6066.

Application Note: This SFR must be claimed in situations where the TOE computes OCSP responses for inclusion in TLS certificate status messages. It may also be claimed if the TOE includes a separate certificate status component (CRL or OCSP provider) and provides an interface to an internal or external OCSP responder that processes certificate status information provided to it by

the TOE. When claimed, certificate status is provided via OCSP stapling contained within TLS Server certificate status message(s).

Evaluation Activities

FCS_OCSPS_EXT.1
For any selection, evaluation activities are included in the TSS, guidance portions, and Tests 2 and 3 within the Test portion of the evaluation activities in FDP_CSI_EXT.1. Additional activities if the first option of FDP_OCSPS_EXT.1.2 is claimed are covered under the evaluation activities for FDP_OCSP_EXT.1.

B.2 Certificate Enrollment

FIA_ESTC_EXT.1 Enrollment over Secure Transport (EST) Client

The inclusion of this selection-based component depends upon selection in FIA_ENR_EXT.1.1.

FIA_ESTC_EXT.1.1

The TSF shall use the Enrollment over Secure Transport (EST) as specified in RFC 7030 to obtain its embedded CA certificate and [**assignment**: *other certificates for the TOE*] from an external certification authority (external CA) associated with an authorized EST server.

FIA_ESTC_EXT.1.2

The TSF shall be able to obtain EST server and CA certificates for authorized EST services via [**selection**:

- *implicit Trust Anchor/Trust Store (TA) configured by **[selection: Security Administrator, CA Operations Staff]** ,*
- *an explicit TA populated via a TLS-authenticated EST CA certificate request in accordance with RFC 7030 section 4.1.2 and FCS_TLSC_EXT.1*

].

FIA_ESTC_EXT.1.3

The TSF shall authenticate EST servers using X.509 certificates that chain to trust store elements from the [**selection**: *implicit Trust Anchor database*, *explicit Trust Anchor/Trust Store*] in accordance with FIA_X509_EXT.1**/Rev** for all EST requests.

FIA_ESTC_EXT.1.4

The TSF shall authenticate its certificate enrollment requests to receive the signing certificate of its embedded CA and [**assignment**: *other certificates required to authenticate the TOE*], from an authorized EST server using [**selection**:

- *HTTP basic authentication transported over TLS in accordance with RFC 7030 section 3.2.3 and FCS_TLSC_EXT.1*,
- *HTTP digest authentication using a cryptographic hash algorithm in accordance with FCS_COP.1**/Hash**, transported over TLS in accordance with RFC 7030 section 3.2.3 and FCS_TLSC_EXT.1*,
- *Certificate-based authentication in accordance with RFC 7030 section 3.3.2 and FCS_TLSC_EXT.2 using [**assignment**: a pre-existing certificate authorized by the EST server]*

].

FIA_ESTC_EXT.1.5

The TSF shall generate authenticated re-enrollment requests in accordance with RFC 7030 Section 3.3.2 and FCS_TLSC_EXT.1, using an existing valid certificate with the same subject name as the requested certificate and which was issued by the external CA.

Application Note: This SFR should be claimed if 'Enrollment over Secure Transport…' is claimed in FIA_ENR_EXT.1.1.

Evaluation Activities

FIA_ESTC_EXT.1
TSS
The evaluator shall examine the TSS to ensure it describes the implementation of this protocol,

the certificates obtained, and any pre-existing certificates or trust anchor databases used by the protocol.
Guidance
The evaluator shall examine the operational guidance to ensure it contains instructions on configuring the TOE so that EST conforms to the description in the TSS.

The evaluator shall examine the operational guidance to ensure it contains instructions for obtaining or configuring the TA database (implicit or explicit) and initial certificates.
Tests
The evaluator shall perform the following tests:

- ***Test 1:*** *The evaluator shall establish an external CA and EST server, and configure the TOE as indicated in the AGD to authorize the EST server for EST services using the external CA. The evaluator shall examine the TOE logs and TA databases using available interfaces to ensure the EST server and external CA's certificates are authorized for EST services.*
- ***Test 2:*** *For each authentication method specified in FIA_ESTC_EXT.1.4, the evaluator shall generate one or more certificate enrollment requests using the authentication method to obtain TOE required certificates from the authorized CA via the EST server established in Test 1. In accordance with guidance documentation, the evaluator shall obtain a sufficient number of certificates in aggregate to allow the TOE to issue certificates to requested servers.*
- ***Test 3:*** *The evaluator shall establish a server with a valid certificate and a monitored client. The evaluator shall configure the TOE so that a TLS session between the monitored client and established through the TOE results in the inspection operation being implemented. The evaluators shall establish a TLS session between the monitored client and the established server through the TOE and observe that the certificate chain returned to the client contains a server certificate issued by the embedded CA's certificate, and the embedded CA's certificate issued by the external CA.*
- ***Test 4:*** *The evaluator shall generate a re-enrollment request and submit it to the authorized EST server in accordance with FIA_ESTC_EXT.1 to update the TOE's embedded CA's signing certificate. The evaluator shall clear any cache, revoke the original CA certificate, and repeat Test 3, observing that the updated certificate for the embedded CA is included in the certificate chain returned to the monitored client.*
- ***Test 5:*** *The evaluator shall establish a second EST server configured to authorize the TOE's EST client but which is not authorized by the client to provide EST services. The evaluator shall generate an enrollment request for the TOE's embedded CA signing certificate, and submit it to the second EST server. The evaluator shall clear any cache and repeat Test 3, observing that the certificate returned by the second EST server is not contained in the certificate chain returned to the monitored client.*

B.3 Inspection Policy Banner

Local policy may require explicit consent to monitoring before inspection of TLS encrypted data. If the STIP may be deployed in an environment where clients might not already have granted this approval, the TOE might be required to obtain this consent. The requirement in this section should be claimed if the TLS session establishment policy requires it.

FTA_TAB.1/TLS TOE Access Banner (Consent to Monitor Banner for TLS Inspection

The inclusion of this selection-based component depends upon selection in FDP_STIP_EXT.1.3.

FTA_TAB.1.1/TLS

Before forwarding decrypted application data intended for the requested server to inspection processing components the TSF shall display **to the monitored client a Security Administrator-specified** advisory **notice and consent** warning message regarding use of the TOE.

Application Note: This SFR should be claimed if 'provide a consent to monitor banner..' is selected in FDP_STIP_EXT.1.3.

Evaluation Activities

FTA_TAB.1/TLS
TSS
The evaluator shall examine the TSS to ensure it details when advisory notice and consent warning messages are used in association with TLS inspection and the circumstances for requiring user consent.
Guidance
The evaluator shall examine the operational guidance to ensure it contains instructions on

configuring the TOE to display consent banners for TLS inspection traffic.
Tests
The evaluator follows the guidance documentation to configure a notice and consent warning message for TLS inspection traffic, and configure rules for displaying the message to monitored clients when requesting TLS sessions to specific servers. The evaluator shall establish a client and server subject to the configured rules, and establish a TLS session through the TOE to the server. The evaluator shall verify that the notice and consent message is displayed.

B.4 Authentication of Monitored Clients

This section describes support for mutual authentication of clients when requested by the TOE to support the following use cases:

- TOE requested mutual authentication: Mutual authentication provides authenticated client attributes that can be used to define exception processing. If the ST claims to support client authentication of monitored clients accessing the TOE, this SFR should be claimed in the selection of FDP_STIP_EXT.1.
- Certificate request from the requested server: Inspection of TLS sessions requiring mutual authentication is a narrow use case for the SSL/TLS inspection proxy, where both the client and server trust the TOE's embedded CA. In such instances, mutual authentication represents an assertion to the requested server that the client has been authenticated. There are other, preferred mechanisms such as SOAP, KERBEROS, XAML assertions, than TLS client authentication using proxy certificates that provide a more accurate representation of the role of a federated identity service provider, in accordance with NIST SP 800-63-03. Also, mutual authentication is typically used to control access to sensitive information authorized only to specific clients, and the willingness of the server's content owner to trust the proxy, providing access such sensitive content further restricts legitimate use cases. Finally, certificates issued to represent users subject to a certificate policy or certificate practice statement, especially those compliant with NIST FIPS 201, may be required to meet an equivalent certificate policy or certificate practice statement.

If mutual authentication by the TOE is to perform the TLS inspection operation on TLS sessions between monitored clients and requested servers requiring mutual authentication, FCS_TTTC_EXT.3, FCS_TTTS_EXT.3, FDP_CER_EXT.4, FDP_CER_EXT.5, FDP_CSI_EXT.2, and FDP_STIP_EXT.2 in this section must be claimed. In addition, the 'mutual authentication inspection' item should be selected in the selection for FDP_TEP_EXT.1.5 and an exception specification to identify servers which are authorized and configured to support mutual authentication inspection must be described in the assignment of FDP_TEP_EXT.1.4. TLS servers requesting mutual authentication are likely to also require revocation information, so it is recommended that FDP_CSIR_EXT.1 selections be made to provide certificate status information, even if the constraint for short validity periods is achieved

FCS_TTTC_EXT.3 Thru-Traffic TLS Inspection Client Protocol with Mutual Authentication Representing Monitored Clients

The inclusion of this selection-based component depends upon selection in FCS_TTTC_EXT.1.1.

FCS_TTTC_EXT.3.1

The TSF shall support mutual authentication using X.509v3 certificates generated in accordance with FDP_CER_EXT.5 for inspection processing operation between a monitored client represented in the generated certificate and a requested server that provides a certificate request in the TLS handshake.

Application Note: This SFR must be claimed if the TSF is capable of inspecting TLS sessions from monitored clients to requested servers requiring client authentication.

Evaluation Activities

FCS_TTTC_EXT.3
TSS
The evaluator shall ensure that the TSS description of the TLS protocol for TLS session establishment includes the use of client-side certificates for TLS mutual authentication to servers when allowed by the configured TLS session establishment policy, described in FDP_TEP_EXT.1.
Guidance
The evaluator shall check the guidance documentation to ensure it contains instructions on configuring the TOE so that the TSF supports inspection of TLS sessions with client authentication. The evaluator shall verify that the AGD guidance provides instructions on how to configure the TLS session establishment policy to identify requested servers it may support for mutual authentication inspection.
Tests
Setup: *The evaluator shall establish one or more monitored clients and one or more requested*

servers that are configured to pass TLS sessions through the TOE and configure the TLS session establishment policy to use the inspection operation for those clients and servers with a supported version and cipher suite. The evaluator shall establish certificates for the servers that are valid in accordance with FIA_X509_EXT.1/STIP and appropriate for the selected cipher suite. For each signature type supported for mutual authentication, the evaluator shall issue a certificate for a monitored client that is valid in accordance with FIA_X509.EXT.1/STIP. The evaluator shall install the appropriate trust anchors within the TSF to validate the client and server certificates. Additional configuration of the requested servers, monitored clients, and TSF are specified in the tests below.

- ***Test 1:*** *The evaluator shall configure the TOE's TLS session establishment policy to allow client authentication to the servers used in this test. For each certificate established for a monitored client, the evaluator shall initiate a TLS session between the monitored client and a requested server configured to require client authentication via a certificate of the indicated type. The evaluator shall then verify that the TLS session between the proxy and the requested server includes a client certificate message containing a certificate issued by the TSF, and that the certificate verifies messages that authenticate the TSF as controlling the private key associated to the certificate.*
- ***Test 2:*** *The evaluator shall configure the TOE's TLS session establishment policy to allow client authentication to the servers used in this test. The evaluator shall configure a requested server to send a certificate request message to clients with a CA field that does not contain the embedded CA of the TOE. The evaluator shall initiate a TLS session from a monitored client using one of the issued certificates for client authentication to the so configured server through the TSF and observe that the TSF does not establish a TLS session with the requested server.*
- ***Test 3:*** *The evaluator shall configure the TOE's TLS session establishment policy to allow client authentication to the servers used in this test. The evaluator shall configure a requested server not to send a certificate request message to clients. The evaluator shall initiate a TLS session from a monitored client using one of the issued certificates for client authentication to the so configured server through the TSF and observe that the TSF does not send a client certificate message or certificate verify message to within its handshake with requested server.*
- ***Test 4:*** *The evaluator shall configure the TOE's TLS session establishment policy to not allow client authentication to the servers used in this test, and if the TSF supports a block-bypass allowance, to block mutual authentication requests to the server. The evaluator shall configure a requested server to send a certificate request message to clients. The evaluator shall initiate a TLS session from a monitored client using one of the issued certificates for client authentication to the so configured server through the TSF and observe that the TSF does not send a client certificate message or certificate verify message to within its handshake with requested server.*
- ***Test 5:*** *(conditional, the TSF supports block-bypass specifications in the TLS session establishment policy): The evaluator shall configure the TOE's TLS session establishment policy to not allow client authentication to the requested server used in this test, and to bypass mutual authentication requests to that server. The evaluator shall configure the requested server to trust the CA used to issue a certificate issued to the monitored client and to send a certificate request messages to clients. The evaluator shall initiate a TLS session from the monitored client using the certificate issued by the CA and trusted by the requested server for client authentication to the so configured server through the TSF. The evaluator shall then observe that the TSF performs the bypass operations and sends a client certificate message containing the certificate established for the monitored client and a certificate verify message that validates the monitored client to the requested server.*

FCS_TTTS_EXT.3 Thru-Traffic TLS Inspection Server Protocol with Mutual Authentication of Monitored Clients

The inclusion of this selection-based component depends upon selection in FCS_TTTS_EXT.1.1.

FCS_TTTS_EXT.3.1

The TSF shall support mutual authentication of TLS clients using X.509v3 certificates.

FCS_TTTS_EXT.3.2

The TSF shall send a Certificate Request message to the TLS client when mutual authentication is required by the configured TLS session establishment policy as defined in FDP_TEP_EXT.1.

FCS_TTTS_EXT.3.3

The TSF shall validate the certificate presented by the client and [**selection**: *allow the connection if the certificate is invalid and an exception is permitted for the client, terminate the connection if the certificate is invalid*].

Application Note: Validity is determined by the identifier verification,

certificate path, the expiration date, and the revocation status in accordance with RFC 5280. Certificate validity is tested in accordance with testing performed for FIA_X509_EXT.1/STIP

FCS_TTTS_EXT.3.4

The TSF shall not establish a TLS session if the distinguished name (DN) or Subject Alternative Name (SAN) contained in a certificate does not match the expected identifier for the client.

Application Note: The client identifier may be in the Subject field or the Subject Alternative Name extension of the certificate.

Evaluation Activities

FCS_TTTS_EXT.3
TSS
The evaluator shall ensure the TSS description of the TLS protocol for TLS session establishment includes the use of client authentication for monitored clients in accordance with the TLS session establishment policy described in FDP_TEP_EXT.1.
Guidance
The evaluator shall check the guidance documentation to ensure it contains instructions on configuring the TOE so that the TSF supports TLS with client authentication for monitored clients. The evaluator shall verify that the AGD guidance provides instructions on how to specify the conditions for when the TSF requests client authentication of monitored clients.
Tests
Setup: *The evaluator shall establish one or more monitored clients and servers that are configured to pass TLS sessions through the TOE, and configure the TLS session establishment policy to require mutual authentication for these clients. The evaluator shall establish certificates for the servers that are valid in accordance with FIA_X509_EXT.1/STIP. Note: Depending on optional features supported by the TOE, it might also be necessary to configure the requested servers to require mutual authentication, and configure the TLS session establishment policy to allow mutual authentication to the requested servers to induce a client certificate request from the TSF.*

- ***Test 1:*** *For each certificate signature algorithm supported, the evaluator shall establish a monitored client with a certificate signed by a trusted CA using the certificate algorithm, where the client properly supports client authentication. For each such client, the evaluator shall initiate a TLS session to a requested server through the TSF. In each case, the evaluator shall observe that valid TLS sessions between the monitored client and the TSF is established and that during the TLS handshake the TSF sends a certificate request message to each monitored client.*
- ***Test 2:*** *The evaluator shall initiate a TLS session between a monitored client and a requested server through the TSF, where the client does not provide a certificate message. The evaluator shall observe that a TLS session between the client and the TSF is not established, and that any TLS session between the TSF and the requested server associated to that TLS session thread is terminated.*
- ***Test 3:*** *The evaluator shall initiate a TLS session between a monitored client and requested server through the TSF, where the monitored client's certificate is issued by a subordinate CA of a trusted root CA and only the root CA is in the TSF trust store. In response to the certificate request, the evaluator shall replace the last byte of the subordinate CA certificate in a valid certificate message from the client to the TSF and send the modified certificate message, along with a valid Certificate Verify message and Finished message to the TSF. The evaluator shall observe TSF logs to verify that the certificate is deemed invalid, that the TSF does not establish a TLS session with the client, and that any TLS session between the TSF and the requested server associated to that TLS session thread is terminated.*
- ***Test 4:*** *The evaluator shall configure the TSF trust store so the root certificate authority that issues the certificate for a monitored client is not trusted. The evaluator shall then initiate a TLS session between a monitored client and requested server using client authentication through the TSF and observe that the TSF does not establish a TLS session with the client and that any TLS session between the TSF and the requested server associated to that TLS session thread is terminated.*
- ***Test 5:*** *The evaluator shall establish a monitored client whose otherwise valid certificate issued by a trusted CA does not include the client authentication purpose in the extended key usage field. The evaluator shall initiate a TLS session between the monitored client and a requested server through the TSF. The evaluator shall observe that the TLS session between the monitored client and the TSF is not established, and that any TLS session between the TSF and the requested server associated to that TLS session thread is terminated.*
- ***Test 6:*** *(conditional, the TLS session establishment supports authenticated attributes of a client in exception specifications): The evaluator shall configure a TLS establishment policy in the TSF to perform mutual authentication for a client. The evaluator shall establish a monitored client subject to the exception specification but having a valid certificate issued by a trusted CA, where the subject identifier and subject alternate name do not match the*

exception specification. The evaluator shall establish a TLS session from the monitored client to a requested server through the TSF and observe that the TLS session between the client and the TSF is not established and any TLS session between the TSF and the requested server associated to that TLS session thread is terminated.

FDP_CER_EXT.4 Certificate Profiles for Client Certificates

The inclusion of this selection-based component depends upon selection in FCS_TTTC_EXT.1.1.

FDP_CER_EXT.4.1

The TSF shall implement a certificate profile function for TLS client certificates issued by a CA embedded within the TOE, and shall ensure that issued certificates are consistent with configured profiles.

Application Note: The CA issuing client certificates may be required to support configured certificate profiles that differ significantly from server certificates, and to support multiple certificate profiles for the clients supported.

FDP_CER_EXT.4.2

The TSF shall generate certificates representing monitored clients using profiles that comply with requirements for certificates as specified in IETF RFC 5280, “Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile” as refined below. At a minimum, the TSF shall ensure that:

a. The version field shall contain the integer 2.
b. The issuerUniqueID or subjectUniqueID fields are not populated.
c. The serialNumber shall be unique with respect to the issuing Certification Authority.
d. The validity field shall specify a notBefore value that does not precede the current time and a notAfter value that does not precede the value specified in notBefore.
e. The issuer field is not empty and is populated with the **[selection: *Security Administrator, CA Operations Staff*]**-configured CA name.
f. The signature field and the algorithm in the subjectPublicKeyInfo field shall contain the OID for a signature algorithm specified in FCS_COP.1/SigGen in the NDcPP.
g. The following extensions are supported:
    a. subjectKeyIdentifier
    b. authorityKeyIdentifier
    c. keyUsage
    d. extendedKeyUsage
    e. certificatePolicy
    f. [**selection**: *basicConstraints*, *cRLDistributionPoints*, *authorityInfoAccess*, *no other extensions*]
h. A subject field containing a null Name (e.g., a sequence of zero relative distinguished names) is accompanied by a populated critical subjectAltName extension.
i. The authorityKeyIdentifier extension in any certificate issued by the TOE must be populated and must be the same as the subjectKeyIdentifier extension contained in the TOE’s embedded CA’s signing certificate
j. Populated keyUsage and extendedKeyUsage fields in the same certificate shall contain consistent values reflecting exclusive TLS server use as follows:

keyUsage	extendedKeyUsage
digitalSignature	clientAuth
digitalSignature, keyEncipherment	clientAuth
digitalSignature,keyAgreement	clientAuth

Application Note: RFC updates to RFC 5280 are included in this requirement. The inclusion of the cRLDistributionPoints and authorityInfoAccess extensions depend on the selections made in FDP_CSIR_EXT.1 and FDP_CSI_EXT.2.3 if claimed.

Uniqueness for the subject key identifier is specific to the instance of the embedded CA. The same configured CA should not issue certificates with different public keys having the same subject key identifier.

FDP_CER_EXT.4.3

The TSF shall implement the following rules for populating certificate fields based on constraints imposed by the TOE's embedded CA's signing certificate:

- The validity field shall specify a notAfter time that does not exceed the notAfter time of the CA's signing certificate.
- The issuer field identifies the [**selection**:
  - *subject,*
  - *[**assignment**: **[selection: Security Administrator, CA Operations Staff]**-assigned identifying information]*

  ] of the CA's signing certificate.
- [**selection**:
  - *The subject name is limited by name constraints specified in the CA's signing certificate,*
  - *[**assignment**: list of rules],*
  - *no other rules*

  ]

FDP_CER_EXT.4.4

The TSF shall implement the following rules for populating certificate fields based on the validated certificate and constraints imposed by the **[selection: *Security Administrator, CA Operations Staff*]**:

a. The Subject/Subject Alternative Name shall be copied from validated client certificate.
b. The notBefore field shall not precede the notBefore field of the validated client certificate.
c. The notAfter field shall not exceed the notAfter field of the validated client certificate.
d. The notAfter field shall not exceed the current time by more than a maximum validity duration value as configured by a **[selection: *Security Administrator, CA Operations Staff*]** user.
e. If the basicConstraints field is configured to be present, it shall be populated with the value cA=False
f. If configured to be present, th epolicy OID and policy mapping fields shall be populated according to [**selection**:
   - *a **[selection: Security Administrator, CA Operations Staff]** configured mapping from validated client certificate values to one or more stated policy OIDs,*
   - *[**assignment**: list of rules]*

   ]

Application Note: Policy OIDs for proxy-issued certificates and mappings to FIPS 201 defined policies may be required to be supported for SSL/TLS inspection proxies issuing certificates representing person-entities subject to FIPS 201 authentication methods, since requested servers requiring client authentication are likely to expect to validate client certificates issued by the equivalent of such a certificate policy. If Policy OIDs are used, the embedded CA may be subject to additional constraints indicated in a Certificate Policy.

Evaluation Activities

FDP_CER_EXT.4
TSS
The evaluator shall examine the TSS to ensure it describes the certificate profile function in accordance with FDP_CER_EXT.4.1 The TSS shall describe how certificate profiles are configured and then selected to issue certificates in accordance with FDP_CER_EXT.4.2. The evaluator shall also ensure that the TSS describes how the TSF ensures that a certificate-requesting subject possesses the applicable private key.
Guidance
The evaluator shall examine the operational guidance to ensure that instructions are available to configure certificate profiles used for certificate generation in accordance with this requirement.
Tests
The evaluator shall perform the following tests:

- ***Test 1:** The evaluator shall configure a certificate profile using the available guidance, and establish a server with a certificate that satisfies FDP_CER_EXT.4.2 items a, b, e, f, h, j, and k, has valid values in all extensions in item g (g.a-g.f), and passes all certificate validation criteria as a TLS server certificate (having extended key usage field of server authentication) in FIA_X509_EXT.1/Rev. The evaluator shall establish a monitored client and request a TLS session to the server through the TOE so that the mutual authentication inspection operation is implemented, and then examine the certificate received at the server from the TOE to ensure it matches the configured certificate profile.*

- ***Test 2:*** *The evaluator shall specifically examine the certificate generated in Test 1 and compare it to both the embedded CA's certificate and the monitored client's certificate to ensure that it satisfies all field constraints in FDP_CER_EXT.4.2, FDP_CER_EXT.4.3, and FDP_CER_EXT.4.4 as configured in the certificate profile.*
- ***Test 3:*** *The evaluator shall conduct the following tests by establishing a monitored client with certificate identical to that used in Test 1, except for the differences described as follows (each in turn). The evaluator shall make any configuration changes to the TOE as indicated, establish a monitored client and submit a TLS request for a requested server requiring mutual authentication through the TOE so that the mutual authentication inspection operation is performed, and observe the certificate received at the requested server has the indicated features:*
  - ***notBefore field test:*** *The evaluator shall assign a notBefore value in the monitored client certificate that precedes both the current time and the value of the notBefore field in the TOE's embedded CA's certificate, and observe the generated certificate has a notBefore value that does not precede the current time.*
  - ***notAfter field test a:*** *The evaluator shall configure the maximum validity duration so that the notAfter value of the TOE's embedded CA certificate does not exceed the current time by more than the maximum validity duration. The evaluator shall assign a notAfter value in the monitored client certificate that exceeds the current time by more than the maximum validity period, and observe that the notAfter field of the generated certificate has a notAfter value that does not exceed the notAfter value of the embedded CA's certificate.*
  - ***notAfter field test b:*** *The evaluator shall configure the maximum validity duration so that the notAfter value in the TOE's embedded CA certificate exceeds the current time by more than maximum validity duration, assign a notAfter value in the monitored client certificate that exceeds the notAfter value in the TOE's embedded CA's certificate, and observe that the notAfter value of the generated certificate does not exceed the current time by more than the maximum validity duration.*
  - ***notAfter field test c:*** *The evaluator shall assign a notAfter value in the monitored client certificate that precedes both the notAfter value in the TOE's embedded CA's certificate, and the current time plus the maximum validity duration, and observe that the generated certificate has a notAfter value that does not exceed the notAfter value of the monitored client's certificate.*
  - ***keyUsage field test:*** *The evaluator shall assign a keyUsage value in the established server certificate that indicates additional usage indicators (e.g., KeyCertSign) and observe that generated certificate has only the Digital Signature and/or Key Encipherment indicators.*
  - ***extendedKeyUsage field test a:*** *The evaluator shall omit the extendedKeyUsage field in the established server certificate and observe that the generated certificate contains the extendedKeyUsage field with value indicating only TLS client authentication.*
  - ***extendedKeyUsage field test b:*** *The evaluator shall populate the extendedKeyUsage field in the established server's certificate to indicate both TLS client authentication and code signing, and observe that the generated certificate only indicates TLS client authentication.*
  - ***extendedKeyUsage field test c:*** *The evaluator shall populate the extendedKeyUsage field in the established server's certificate to indicate any usage, and observe the generated certificate only indicates TLS client authentication.*

FDP_CER_EXT.5 Certificate Issuance Rules for Client Certificates

The inclusion of this selection-based component depends upon selection in FCS_TTTC_EXT.1.1.

FDP_CER_EXT.5.1

The TSF shall issue certificates in response to a validated client certificate according to the following rules: The issued certificate is in compliance with a current certificate profile defined in accordance with FDP_CER_EXT.5 and

- The TLS session establishment policy is configured to allow inspection of TLS sessions with mutual authentication between the monitored client whose certificate is validated by the TSF and one or more requested servers,
- The specific requested server includes a Certificate Request message in the TLS handshake,

[**selection**:

- *A valid certificate for the same subject is not present in cache,*
- *The embedded CA certificate's name space allows issuance of a certificate that represents the authenticated client,*
- *No other constraints*

].

Application Note: Caching client certificates is neither required nor preferred, since such storage of private keys associated to signature keys is strictly controlled by various certificate policies and practice statements, especially when the validated client certificate associated to the monitored client is issued under NIST FIPS 201. If supported, the time in cache for client certificates should be limited to the minimal revocation time (emergency revocation) allowed for validated certificates used by any monitored client.

FDP_CER_EXT.5.2

The TSF shall reject all certificate requests originating external to the TOE.

Evaluation Activities

FDP_CER_EXT.5
TSS
The evaluator shall examine the TSS to ensure it describes the certificate issuance rules, and verify that any interfaces available for external certificate requests (CMC, EST, PKCS#10 or any other request format) are identified.
Guidance
The evaluator shall examine the operational guidance to ensure that it contains instructions for any configuration aspects of any certificate issuance approval function and the steps needed to prevent receipt and approval of external requests.
Tests
The evaluator shall generate certificates that originate external to the TOE and verify that they are rejected.

FDP_CSI_EXT.2 Certificate Status Information for Client Certificates

The inclusion of this selection-based component depends upon selection in FCS_TTTC_EXT.1.1, FDP_CSIR_EXT.1.1.

FDP_CSI_EXT.2.1

The TSF shall generate certificate status information for issued client certificates whose format complies with [**selection**: *ITU-T Recommendation X.509v2 CRL, the OCSP standard as defined by RFC 6960*].

FDP_CSI_EXT.2.2

The TSF shall support changes to the status of a certificate in accordance with the following rules:

- as directed by **[selection: *Security Administrator, CA Operations Staff*]** and
- [**selection**:
  - *a certificate in cache is revoked when a certificate representing the same subject is received for client authentication and either*
    - *the validation of the received certificate fails, or*
    - *the validation of the received certificate passes and the certificate fields of the validated certificate would result in a different certificate being issued under the current profile in accordance with FDP_CER_EXT.4*

    ,
  - *[**assignment**: other rules for revocation of issued certificates],*
  - *no other rules*

  ]

Application Note: In order to meet revocation requirements associated with credentials issued under FIPS 201, the first item of the second selection in this element must be claimed if cache is provided. Automated rules for revoking certificates in response to the TOE's discovery that a previously issued certificate is no longer appropriate for the subject, or due to cache clearing, timeouts, or other rules should be described in the assignment of FDP_CSI_EXT.2.2.

FDP_CSI_EXT.2.3

The TSF shall [**selection**: *provide, interface with the Operational Environment to provide*] certificate status information generated in accordance with FDP_CSI_EXT.2.1 via [**selection**: *posting CRLs at the location specified in the cRLDistributionPoints of the issued certificate, an OCSP mechanism indicated in the authorityInfoAccess extension of the issued certificate*].

Application Note: Based on the selection, the ST author must choose the

appropriate requirements from Appendix B.1 of this PP-Module.

The ST should specify the format(s) used to supply certificate status information in FDP_CSI_EXT.2.1, and the mechanism(s) used to provide the status to relying parties including all servers authorized to use mutually authenticated TLS in accordance with the configured TLS session establishment policy, identified in FDP_TEP_EXT.1, in the second selection in FDP_CSI_EXT.2.3. If CRLs are identified in FDP_CSI_EXT.2.1, then cRLDistributionPoints must be claimed in FDP_CSI_EXT.2.3 and in FDP_CER_EXT.4.2 item (g), sub-item (f). If the OCSP standard is selected in FDP_CSI_EXT.2.1, then 'authorityInfoAccess' must be selected in the second selection of FDP_CSI_EXT.2.3 and in FDP_CER_EXT.4.2 item (g), sub-item (f).

Evaluation Activities

FDP_CSI_EXT.2
TSS
The evaluator shall examine the TSS to ensure it describes the certificate status function and applicable formats, in accordance with this requirement, that can be used to issue certificate status. The TSS must reflect the selection made by the ST author as well as the selection-based requirements from Appendix B.1 of the STIP PP-Module for CRL or OCSP information. The evaluator shall also ensure that the TSS describes the process for approving changes to the status of a certificate, including the interfaces that must be used.
Guidance
The evaluator shall examine the operational guidance to ensure that it contains instructions for any configuration aspects of the certificate status change function and the steps needed to perform an approval, as well as any configuration required for interfaces to external certificate status providers.
Tests
Based on the selections, the evaluator shall perform the following tests. It is recommended that these be performed in conjunction with applicable tests associated with the requirements claimed in Appendix B.1:

- ***Test 1:*** *For each certificate status format identified in FCS_CSI_EXT.2.1, the evaluator shall cause a valid client certificate to be issued by the TOE. The evaluator shall then cause the TOE to issue certificate status information. The evaluator shall check the certificate status information using all indicated methods identified in FCS_CSI_EXT.2.3 to verify that each reflects that the certificate is valid.*
- ***Test 2:*** *For each selected certificate status format (CRLv2 or OCSP) identified in FCS_CSI_EXT.2.1, and for each mechanism indicated in FDP_CSI_EXT.2.2, the evaluator shall cause a valid client certificate from the TOE to be revoked. The evaluator shall then cause the TOE to issue certificate status information. The evaluator shall check the certificate status information using all methods (cRLDistributionPoints, authorityInfoAccess, or OCSP Stapling) identified in FCS_CSI_EXT.2.3 to verify that each method reflects that the certificate is revoked.*

FDP_STIP_EXT.2 Mutual Authentication Inspection Operation

The inclusion of this selection-based component depends upon selection in FCS_TTTC_EXT.1.1.

FDP_STIP_EXT.2.1

The TSF shall be capable of providing mutual authentication of the monitored client to a requested server when performing the inspection operation when mutual authentication is allowed for the requested server by the configured policy, and the TLS handshake with the requested server includes a certificate request.

Application Note: The policy is flexible; it could be static policy or a policy associated with certain clients, servers, or connections.

FDP_STIP_EXT.2.2

After receiving the TLS client certificate from the monitored client, the TSF shall be able to generate a certificate representing the client in accordance with FDP_CER_EXT.5 and [**selection**: *obtain a valid certificate representing the client from cache*, *no other method*] matching the current certificate profile.

Application Note: Certificate caching of client certificates is not required. However, in the case where certificate caching is supported, the TSF will still need to perform certificate generation if the cached certificate does not match the current profile determined by FDP_CER_EXT.4 which depends on values derived from the certificate provided by the monitored client.

FDP_STIP_EXT.2.3

After obtaining a certificate representing the monitored client, the TSF shall send the client certificate and certificate verify messages to the requested server.

Application Note: This element completes the TLS handshake between the TOE and the requested server as a complete TLS handshake with mutual authentication.

Evaluation Activities

FDP_STIP_EXT.2
TSS
The evaluator shall examine the TSS to ensure that inspection of mutually authenticated TLS sessions is described and meets the requirements of FDP_STIP_EXT.2.

If the selection in FDP_STIP_EXT.2.2 indicates client certificate caching is supported, the evaluator shall examine the TSS to ensure that the cache is described, as well as the mechanism to determine when certificates are cached and when new certificates are obtained.

The evaluator shall examine the TSS and confirm that the TSF only sends a TSF-generated certificate message and certificate validate message to a requested server matching an exception specification after it verifies that the certificate meets the configured certificate profile associated to a validated client certificate received from the monitored client requesting TLS to the server
Guidance
The evaluator shall examine operational documentation and verify that instructions to configure the mutual authentication inspection operation is provided.
Tests
Setup: *The evaluator shall follow AGD guidance to configure the TSF. The evaluator shall establish a monitored client able to initiate a TLS session compliant with FCS_TLSC_EXT.2 and which is issued a certificate compliant with FIA_X509_EXT.1/STIP for client authentication from a trusted CA that is different than the embedded CA of the TOE.*

The evaluator shall ensure the validity of the client's certificate is short enough to accommodate Test 3 below. The evaluator shall establish a server able to establish TLS sessions in accordance with FCS_TLSS_EXT.2 and configured to support mutual authentication of the client and which is configured to trust the TOE's embedded CA.

The evaluator shall ensure the server is issued a certificate issued by a trusted CA that is different than the TOE's embedded CA, and which is valid in accordance with FIA_X509_EXT.1/STIP for server authentication.

The evaluator shall follow AGD guidance to configure the TLS session establishment policy so mutually authenticated TLS sessions through the TOE between the client and server is allowed and will be inspected.

The evaluator shall use appropriate tools to monitor the traffic between the clients and the TOE, and between the TOE and the server to observe the TLS handshake messages. The evaluator shall perform the following tests in order:

- ***Test 1:*** *The evaluator shall configure the server to not require mutual authentication of the client, initiate a TLS session to the server through the TSF, and observe a TLS session between the TSF and the server is established and does not include a certificate message or certificate verify message from the TSF. The evaluator shall inspect the server certificate received at the client to confirm that it was issued by the embedded CA of the TOE, confirming that the inspection operation was implemented. The evaluator shall inspect the certificate repository of the TOE and confirm that no certificate representing the client is present.*
- ***Test 2:*** *The evaluator shall configure the server to require mutual authentication of the client, initiate a TLS session from the client to the requested server, and observe the traffic between the TOE and the server to verify that the TSF sends a certificate message containing a certificate issued by the embedded CA of the TOE, and a certificate verify message that validates the TSF's possession of the corresponding private key. The evaluator shall examine the certificate repository of the TOE to confirm that the certificate observed in the certificate message is present in the repository.*
- ***Test 3:*** *Adjusting the time of the TSF if necessary so that the initial certificate issued to the client is expired, the evaluator shall establish a new certificate for the client, using the same subject but using a validity period that is valid in the current time setting. The evaluator shall initiate a TLS session between the client and the server requiring mutual authentication through the TOE and observe that a TLS session containing a certificate message with a new certificate generated by the embedded CA of the TOE, and a certificate verify message that validates the TSF's possession of the associated private key. The evaluator shall examine the certificate repository of the TOE and verify that both certificates representing the client are present.*
- ***Test 4:*** *The evaluator shall initiate a new TLS session between the client and server through the TOE, where the certificate in the certificate message from the client is modified*

in the last byte, and a valid certificate verify message is sent for the unmodified certificate. The evaluator shall observe that a TLS session between the client and the TSF is not established, and that any TLS session between the TSF and the server associated to that TLS session thread is terminated.

B.5 Other Selection-Based SFRs

FAU_SCR_EXT.1 Certificate Repository Review

The inclusion of this selection-based component depends upon selection in FAU_GCR_EXT.1.1.

FAU_SCR_EXT.1.1

The TSF shall [**selection**: *provide*, *invoke the Operational Environment to provide*] the ability to search certificates containing specified values of the following certificate fields: **[selection:**

- ***subject name,***
- ***individual components of Subject Alternative Name,***
- ***subject ID,***
- ***issuer ID,***
- ***algorithm ID,***
- ***public key,***
- ***key usage,***
- ***extended key usage,***
- ***serial number,***
- ***[assignment: list of other certificate fields]***

] returning all matching certificates and [**assignment**: *object identifier(s)*] of matching certificate.

Application Note: This SFR must be claimed if the selection in FAU_GCR_EXT.1.1 is 'store.' It may be claimed if the selection in FAU_GCR_EXT.1.1 is 'invokes the Operational Environment to store' when the TSF provides an interface to the certificate repository to perform searches. The ability to search on certificate fields is useful for conducting forensic analysis. If the certificate repository is stored within the TOE boundary, then the first item of the first selection is chosen. If the repository is stored in the OE, but the auditor uses TSF interfaces to perform this function on the repository, then the second item of the first selection is chosen. It is allowed that this function be provided entirely by the OE (when the repository is stored in the OE); if this is the case, then this requirement is not included in the ST, but instead the OE.CERTIFICATE_REPOSITORY_SEARCH objective is included (this objective is omitted in the other two cases, when this SFR is included in the ST).

In the second selection and assignment, the ST author includes/fills in the values that can be searched on for this function; at least one value is required to be selected.

Evaluation Activities

FAU_SCR_EXT.1
TSS
The evaluator shall examine the TSS to ensure it describes the certificate repository if the TSF stores it, or describes the interfaces to the operational environment if the certificate repository is stored external to the TOE. The evaluator shall check the TSS to ensure it describes how to search the certificate repository for the selected items.
Guidance
The evaluator shall examine the operational guidance to ensure it contains instructions for searching the specified information.
Tests
The following test applies regardless of the selection made in the first selection in the SFR. The test activities can be conducted in conjunction with those for FDP_CER_EXT.1 and FAU_GCR_EXT.1.

The evaluator shall generate a sufficient number and variety of certificates to populate the repository with certificates having at least two values for each of the search fields selected in this SFR. The evaluator shall then—following the instructions in the operational guidance—search the repository or audit record for certificates containing specific values for each search field included in the ST, and confirm all certificates matching the search criteria are returned, that all returned certificates match the criteria, and that the object identifier for each matched

item is returned. The evaluator shall confirm that the object identifier returned matches the audit events associated with generation of the certificates in accordance with FAU_GEN.1.

FCS_CKM_EXT.5 Public Key Integrity

The inclusion of this selection-based component depends upon selection in FDP_STG_EXT.1.1.

FCS_CKM_EXT.5.1

The TSF shall protect persistent public keys against undetected modification through the use of [**selection**: *digital signatures* ***(in accordance with FCS_COP.1/SigGen)***, *keyed hashes* ***(in accordance with FCS_COP.1/KeyedHash)***].

FCS_CKM_EXT.5.2

The [**selection**: *digital signature*, *keyed hash*] used to protect a public key shall be verified upon [**assignment**: *criteria for automated verification*].

Application Note: This SFR is included when the second selection in FDP_STG_EXT.1.1 is chosen, and applies to the public keys listed in that SFR.

The selections in FCS_CKM_EXT.5.1 and FCS_CKM_EXT.5.2 should agree, and the assignment in FCS_CKM.5.2 for the criteria for automated verification can be event or time based and should provide operationally relevant integrity failure detection, for which recovery is feasible.

Evaluation Activities

FCS_CKM_EXT.5
TSS
The evaluator shall examine the TSS to ensure it describes each applicable public key, where it is stored and protected, the purpose of the public key, the mechanism used to protect the public key from undetected modification, and the method (for each public key) by which the integrity of the key is checked in accordance with FCS_CKM_EXT.5.2.
Guidance
There are no guidance EAs for this component.
Tests
NOTE: It might not be possible to access public keys via the TOE interface. If that is the case, then the evaluator must describe the interface and indicate why the interface does not allow access to the public keys.

For each public key identified in the TSS, the evaluator shall perform the following test:

The evaluator shall perform an action to invalidate the integrity of each public key and then verify that the TSF detects the invalid key.

FCS_TTTC_EXT.4 STIP Client-Side Support for Renegotiation

The inclusion of this selection-based component depends upon selection in FCS_TTTC_EXT.1.1.

FCS_TTTC_EXT.4.1

The TSF shall support secure renegotiation on STIP TLS connections through use of the “renegotiation_info” TLS extension in accordance with RFC 5746.

FCS_TTTC_EXT.4.2

The TSF shall include [**selection**: *renegotiation_info extension*, *TLS_EMPTY_RENEGOTIATION_INFO_SCSV cipher suite*] in the Client Hello message.

Application Note: This SFR is included when “session renegotiation” in FCS_TTTC_EXT.1.1 is chosen. RFC 5746 defines an extension to TLS that binds renegotiation handshakes to the cryptography in the original handshake. The cipher suite included in the selection is a means for clients to be compatible with servers that don’t support the extension. It is recommended that client implementations support both the cipher suite and the extension.

FCS_TTTC_EXT.4.3

The TSF shall ensure that renegotiation is performed before [**selection**: *[**assignment**: renegotiation rules], 2^20 64-bit data blocks are encrypted using TDES cipher suites using the same key*].

Application Note: If a TDES cipher suite is selected in FCS_TTTC_EXT.1.1, the amount of data encrypted with the same key is limited in accordance with NIST SP800-67R2, section 3.4, and the second selection should be chosen.

Evaluation Activities

FCS_TTTC_EXT.4
TSS
The evaluator shall examine the TSS to validate that it describes the method used to support renegotiation.
Guidance
There are no guidance EAs for this component beyond what is specified for FCS_TTTC_EXT.4.3.
Tests
The evaluator shall perform the following tests:

- ***Test 1:*** *The evaluator shall use a network packet analyzer and/or sniffer to capture the traffic between the TSF and a requested server during inspection of a TLS session between a monitored client and the requested server through the TOE. The evaluator shall verify that either the renegotiation_info field or the SCSV cipher suite is included in the Client Hello message during the initial handshake.*
- ***Test 2:*** *The evaluator shall verify the TSF's handling of Server Hello messages received from a requested server during an authorized inspection of a TLS session between a monitored client and the requested server through the TOE, during the initial handshake that include the renegotiation_info extension. The evaluator shall modify the length portion of this field in the Server Hello message to be non-zero and verify that the TSF sends a failure and terminates the connection. The evaluator shall verify that a properly formatted field during an authorized inspection of traffic to the server results in a successful TLS connection between the TSF and the requested server.*
- ***Test 3:*** *The evaluator shall cause the TSF to initiate renegotiation with the requested server and verify that the Client Hello message received by the requested server containsthe renegotiation_info extension. The evaluator shall cause the requested server to send a Server Hello message with a renegotiation info extension containing data in which one or both of the client_verify_data or server_verify_data value is modified. The evaluator shall verify that the TSF terminates the connection.*

FCS_TTTC_EXT.4.3
TSS
The evaluator shall verify that the TSS describes the mechanisms used to specify when renegotiation occurs.
Guidance
The evaluator shall check the AGD guidance documentation to ensure that instructions for any configurable features of the TLS implementation required to meet this requirement are provided.
Tests
The evaluator shall perform the following tests:

- ***Test 1:*** *For any mechanism specified, the evaluator will establish one or more monitored client and requested servers configured to use each of the supported cipher suites through the TSF. For each supported cipher suite, the evaluator shall initiate a session between the monitored client to a requested server and observe network traffic between the TOE and the requested server to confirm that the indicated cipher suite is negotiated successfully. The evaluator shall then send application data over the inspected channel between the monitored client until the renegotiation criteria is met. The evaluator shall observe that the TSF terminates or renegotiates the TLS session as specified by the renegotiation mechanism.*
- ***Test 2:*** *(conditional, the ST selects "2^20 64-bit data blocks are encrypted using TDES cipher suites using the same key" in FCS_TTTC_EXT.4.3): the evaluator shall establish one or more monitored client and requested servers configured to use each of the supported cipher suites using TDES through the TSF. For each supported cipher suite using TDES, the evaluator shall configure the TLS session establishment of the TSF to inspect such traffic, to include setting appropriate exception specifications. The evaluator shall initiate a session between the monitored client to a requested server and observe network traffic between the TOE and the requested server to confirm that the indicated cipher suite using TDES is negotiated successfully. The evaluator shall then send application data over the inspected channel between the monitored client and the requested server so that the number of data blocks encrypted under TDES will exceed 2^20. The evaluator shall observe that the TSF terminates or renegotiates the TLS session before the number of data blocks encrypted to the requested server exceeds 2^20.*

FCS_TTTS_EXT.4 STIP Server-Side Support for Renegotiation

The inclusion of this selection-based component depends upon selection in FCS_TTTS_EXT.1.1.

FCS_TTTS_EXT.4.1

The TSF shall support the "renegotiation_info" TLS extension in accordance with RFC 5746.

FCS_TTTS_EXT.4.2

The TSF shall include the renegotiation_info extension in Server Hello messages.

Application Note: This SFR is included when “session renegotiation” in FCS_TTTS_EXT.1.1 is chosen. RFC 5746 defines an extension to TLS that binds renegotiation handshakes to the cryptography in the original handshake.

Evaluation Activities

FCS_TTTS_EXT.4
TSS
The evaluator shall examine the TSS to validate it describes the method used to support renegotiation.
Guidance
There are no guidance EAs for this component.
Tests
The evaluator shall establish a monitored client that supports secure renegotiation and the renegotiation_info extension, and a requested server that is authorized for the inspection operation. The evaluator shall then perform the following tests:

- ***Test 1:*** *The evaluator shall use a network packet analyzer or sniffer to capture the traffic between the TSF and a monitored client. The evaluator shall initiate a TLS session between the monitored client and the requested server through the TOE, and verify the renegotiation_info field is included in the Server Hello message sent from the TSF to the monitored client.*
- ***Test 2:*** *The evaluator shall initiate a new (initial) TLS session between the monitored client and the requested server through the TOE, where the Client Hello message includes a renegotiation_info extension with non-zero length, and verify the TSF sends a failure and terminates the connection. The evaluator shall verify that a properly formatted field results in a successful TLS connection.*
- ***Test 3:*** *The evaluator shall send a renegotiation request from the monitored client to the TSF containing a modified client_verify_data value in the Client Hello message. The evaluator shall verify the TSF terminates the connection.*

Appendix C - Extended Component Definitions

This appendix contains the definitions for all extended requirements specified in the PP-Module.

C.1 Extended Components Table

All extended components specified in the PP-Module are listed in this table:

Table 5: Extended Component Definitions

Functional Class	Functional Components
Security Audit (FAU)	FAU_GCR_EXT Generation of Certificate Repository
Other Selection-Based SFRs	FAU_SCR_EXT Certificate Repository Review

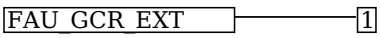
C.2 Extended Component Definitions

C.2.1 FAU_GCR_EXT Generation of Certificate Repository

Family Behavior

Components in this family define requirements for persistent certificate storage in a repository.

Component Leveling



FAU_GCR_EXT.1, Generation of Certificate Repository, requires a conformant TOE to specify how it stores certificates that are issued by the TSF.

Management: FAU_GCR_EXT.1

No specific management functions are identified.

Audit: FAU_GCR_EXT.1

There are no auditable events foreseen.

FAU_GCR_EXT.1 Generation of Certificate Repository

Hierarchical to: No other components.

Dependencies to: FDP_CER_EXT.1 Certificate Profiles for Server Certificates

FDP_CER_EXT.3 Certificate Issuance Rules for Server Certificates

FAU_GCR_EXT.1.1

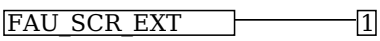
The TSF shall [**selection**: *store*, *invoke the Operational Environment to store*] certificates issued by the TSF.

C.2.2 FAU_SCR_EXT Certificate Repository Review

Family Behavior

Components in this family define requirements for searching the contents of a certificate repository.

Component Leveling



FAU_SCR_EXT.1, Certificate Repository Review, requires a conformant TOE to support the searching of a certificate repository based on the values of specific certificate fields.

Management: FAU_SCR_EXT.1

The following actions could be considered for the management functions in FMT:

- Ability to search the certificate repository

Audit: FAU_SCR_EXT.1

There are no auditable events foreseen.

FAU_SCR_EXT.1 Certificate Repository Review

Hierarchical to: No other components.

Dependencies to: FAU_GCR_EXT.1 Generation of Certificate Repository

FAU_SCR_EXT.1.1

The TSF shall [**selection**: *provide, invoke the Operational Environment to provide*] the ability to search certificates containing specified values of the following certificate fields: [**assignment**: *list of certificate fields*], returning all matching certificates and [**assignment**: *object identifier(s)*] of matching certificates.

Appendix D - Implicitly Satisfied Requirements

This appendix lists requirements that should be considered satisfied by products successfully evaluated against this PP. These requirements are not featured explicitly as SFRs and should not be included in the ST. They are not included as standalone SFRs because it would increase the time, cost, and complexity of evaluation. This approach is permitted by [CC] Part 1, 8.2 Dependencies between components.

This information benefits systems engineering activities which call for inclusion of particular security controls. Evaluation against the PP provides evidence that these controls are present and have been evaluated.

. **Table 6: Implicitly Satisfied Requirements**

Requirement	Rationale for Satisfaction
FCS_CKM.2 – Cryptographic Key Distribution, or FCS_COP.1 – Cryptographic Operation	FCS_CKM.1 (which is defined in this PP-Module as FCS_CKM.1/VPN) requires one of FCS_CKM.2 or FCS_COP.1 to be claimed so that the generated keys can serve some security-relevant purpose. Each of the Base-PPs for this PPModule define an iteration of FCS_COP.1 for symmetric cryptography that is expected to use the IKE keys generated by FCS_CKM.1/VPN. Therefore, this dependency is satisfied through requirements defined in the Base-PPs.

Appendix E - Entropy Documentation and Assessment

The TOE does not require any additional supplementary information to describe its entropy source(s) beyond the requirements outlined in the Base-PP. As with other Base-PP requirements, the only additional requirement is that the entropy documentation also applies to the specific STIP capabilities of the TOE in addition to the functionality required by the claimed Base-PP.

Appendix F - Acronyms

Acronym	Meaning
Base-PP	Base Protection Profile
CA	Certificate Authority
CC	Common Criteria
CEM	Common Evaluation Methodology
HTTP	HyperText Transfer Protocol
OE	Operational Environment
PP	Protection Profile
PP-Configuration	Protection Profile Configuration
PP-Module	Protection Profile Module
SAR	Security Assurance Requirement
SFR	Security Functional Requirement
SSL/TLS	Secure Sockets Layer/Transport Layer Security
ST	Security Target
STIP	SSL/TLS Inspection Proxy
TA	Trust Anchor (Trust Store)
TOE	Target of Evaluation
TSF	TOE Security Functionality
TSFI	TSF Interface
TSS	TOE Summary Specification
URL	Uniform Resource Locator

Appendix G - Bibliography

Identifier	Title
[CC]	Common Criteria for Information Technology Security Evaluation - • Part 1: Introduction and General Model, CCMB-2017-04-001, Version 3.1 Revision 5, April 2017. • Part 2: Security Functional Components, CCMB-2017-04-002, Version 3.1 Revision 5, April 2017. • Part 3: Security Assurance Components, CCMB-2017-04-003, Version 3.1 Revision 5, April 2017.
[ND-SD]	Supporting Document - Mandatory Technical Document - Evaluation Activities for Network Device cPP, Version 2.2, December 2019
[NDcPP]	collaborative Protection Profile for Network Devices, Version 2.2E, March 2020