collaborative Protection Profile for Stateful Traffic Filter Firewalls

Version 0.1

05-Sep-2014

Acknowledgements

This collaborative Protection Profile (cPP) was developed by the Network international Technical Community with representatives from industry, Government agencies, Common Criteria Test Laboratories, and members of academia.

0. Preface

0.1 Objectives of Document

This document presents the Common Criteria (CC) collaborative Protection Profile (cPP) to express the security functional requirements (SFRs) and security assurance requirements (SARs) for a Stateful Traffic Filter Firewall. The Evaluation Activities that specify the actions the evaluator performs to determine if a product satisfies the SFRs captured within this cPP are described in [SD-ND] and [SD-FW].

0.2 Scope of Document

The scope of the cPP within the development and evaluation process is described in the Common Criteria for Information Technology Security Evaluation [CC]. In particular, a cPP defines the IT security requirements of a generic type of TOE and specifies the functional and assurance security measures to be offered by that TOE to meet stated requirements [CC1, Section C.1].

0.3 Intended Readership

The target audiences of this cPP are developers, CC consumers, system integrators, evaluators and schemes.

0.4 Related Documents

Common Criteria[1]

[CC1] Common Criteria for Information Technology Security Evaluation,
Part 1: Introduction and General Model,
CCMB-2012-09-001, Version 3.1 Revision 4, September 2012.

[CC2] Common Criteria for Information Technology Security Evaluation,
Part 2: Security Functional Components,
CCMB-2012-09-002, Version 3.1 Revision 4, September 2012.

[CC3] Common Criteria for Information Technology Security Evaluation,
Part 3: Security Assurance Components,
CCMB-2012-09-003, Version 3.1 Revision 4, September 2012.

[CEM] Common Methodology for Information Technology Security Evaluation,
Evaluation Methodology,
CCMB-2012-09-004, Version 3.1, Revision 4, September 2012.

[1] For details see http://www.commoncriteriaportal.org/

Other Documents

[SD-FW] Evaluation Activities for Stateful Traffic Filter Firewalls cPP, Version 0.1, September 2014

[SD-ND] **Evaluation Activities for Network Device cPP, Version 0.1, September 2014**

0.5 Revision History

Version	Date	Description
0.1	05-Sep-2014	Draft published for Public review

Contents

Acknowledgements 2
0. Preface 3
0.1 Objectives of Document 3
0.2 Scope of Document 3
0.3 Intended Readership 3
0.4 Related Documents 3
0.5 Revision History 5
1. PP Introduction 11
1.1 PP Reference Identification 11
1.2 TOE Overview 11
1.3 TOE Usage 11
2. CC Conformance 11
3. Security Problem Definition 13
3.1 Threats 13
3.1.1 Communications with the Network Device 13
3.1.1.1 T.UNAUTHORIZED_ADMINISTRATOR_ACCESS 14
3.1.1.2 T.WEAK_CRYPTOGRAPHY 14
3.1.1.3 T.UNTRUSTED_COMMUNICATION_CHANNELS 14
3.1.1.4 T.WEAK_AUTHENTICATION_ENDPOINTS 14
3.1.2 Valid Updates 14
3.1.2.1 T.UPDATE_COMPROMISE 14
3.1.3 Audited Activity 15
3.1.3.1 T.UNDETECTED_ACTIVITY 15
3.1.4 Administrator and Device Credentials and Data 15
3.1.4.1 T.SECURITY_FUNCTIONALITY_COMPROMISE 15
3.1.4.2 T.PASSWORD_CRACKING 16
3.1.5 Device Failure 16
3.1.5.1 T.SECURITY_FUNCTIONALITY_FAILURE 16
3.2 Assumptions 18
3.2.1 A.PHYSICAL_PROTECTION 18
3.2.2 A.LIMITED_FUNCTIONALITY 19
3.2.3 A.NO_THRU_TRAFFIC_PROTECTION ..... **Error! Bookmark not defined.**
3.2.4 A.TRUSTED_ADMINSTRATOR 19
3.2.5 A.REGULAR_UPDATES 19
3.2.6 A.ADMIN_CREDENTIALS_SECURE 19
3.3 Organizational Security Policy 19
3.3.1 P.ACCESS_BANNER 19
4. Security Objectives 20
4.1 Security Objectives for the Operational Environment 20
4.1.1 OE.NO_GENERAL_PURPOSE 20
4.1.2 OE.PHYSICAL 20
4.1.3 OE.TRUSTED_ADMIN 20
5. Security Functional Requirements 21
5.1 Conventions 21
5.2 Security Audit (FAU) 21
5.2.1 Security Audit Data generation (FAU_GEN) 21
5.2.1.1 FAU_GEN.1 Audit data generation 22
5.2.1.2 FAU_GEN.2 User identity association 25
5.2.2 Security audit event storage (Extended – FAU_STG_EXT) 26
5.2.2.1 FAU_ STG_EXT.1 External Audit Trail Storage 26
5.3 Cryptographic Support (FCS) 26
5.3.1 Cryptographic Key Management (FCS_CKM) 26
5.3.1.1 FCS_CKM.1 Cryptographic Key Generation 26
5.3.1.2 FCS_CKM.2 Cryptographic Key Establishment 27
5.3.1.3 FCS_CKM.4 Cryptographic Key Destruction 28

5.3.2 Cryptographic Operation (FCS_COP) ... 29
5.3.2.1 FCS_COP.1 Cryptographic Operation ... 29
5.3.3 Random Bit Generation (Extended – FCS_RBG_EXT) ... 30
5.3.3.1 FCS_RBG_EXT.1 Random Bit Generation ... 30
5.4 User Data Protection (FDP) ... 31
5.4.1 Residual information protection (FDP_RIP) ... 31
5.4.1.1 FDP_RIP.2 Full Residual Information Protection ... 31
5.5 Identification and Authentication (FIA) ... 32
5.5.1 Password Management (Extended – FIA_PMG_EXT) ... 32
5.5.1.1 FIA_PMG_EXT.1 Password Management ... 32
5.5.2 User Identification and Authentication (Extended – FIA_UIA_EXT) ... 32
5.5.2.1 FIA_UIA_EXT.1 User Identification and Authentication ... 32
5.5.3 User authentication (FIA_UAU) (Extended – FIA_UAU_EXT) ... 33
5.5.3.1 FIA_UAU_EXT.2 Password-based Authentication Mechanism ... 33
5.5.3.2 FIA_UAU.7 Protected Authentication Feedback ... 33
5.5.4 Authentication using X.509 certificates (Extended – FIA_X509_EXT) ... 33
5.5.4.1 FIA_X509_EXT.1 X.509 Certificate Validation ... 33
5.5.4.2 FIA_X509_EXT.2 X.509 Certificate Authentication ... 34
5.5.4.3 FIA_X509_EXT.3 X.509 Certificate Requests ... 35
5.6 Security Management (FMT) ... 35
5.6.1 Management of functions in TSF (FMT_MOF) ... 35
5.6.2 Management of TSF Data (FMT_MTD) ... 36
5.6.2.1 FMT_MTD.1 Management of TSF Data ... 36
5.6.3 Specification of Management Functions (FMT_SMF) ... 36
5.6.3.1 FMT_SMF.1 Specification of Management Functions ... 36
5.6.4 Security management roles (FMT_SMR) ... 37
5.6.4.1 FMT_SMR.2 Restrictions on security roles ... 37
5.7 Protection of the TSF (FPT) ... 37
5.7.1 Internal TOE TSF data transfer (FPT_ITT) ... 37
5.7.1.1 FPT_ITT.1 Basic Internal TSF Data Transfer Protection (Refinement) ... 37
5.7.2 Protection of TSF Data (Extended – FPT_SKP_EXT) ... 38
5.7.2.1 FPT_SKP_EXT.1 Protection of TSF Data (for reading of all symmetric keys) .. 38
5.7.3 Protection of Administrator Passwords (Extended – FPT_APW_EXT) ... 38
5.7.3.1 FPT_APW_EXT.1 Protection of Administrator Passwords ... 38
5.7.4 TSF testing (Extended) ... 38
5.7.4.1 TSF Testing (Extended) ... 39
5.7.5 Trusted Update (FPT_TUD_EXT) ... 39
5.7.5.1 FPT_TUD_EXT.1 Trusted Update ... 39
5.7.6 Time stamps (FPT_STM) ... 40
5.7.6.1 FPT_STM.1 Reliable Time Stamps ... 40
5.8 TOE Access (FTA) ... 41
5.8.1 TSF-initiated Session Locking (Extended – FTA_SSL_EXT) ... 41
5.8.1.1 FTA_SSL_EXT.1 TSF-initiated Session Locking ... 41
5.8.2 Session locking and termination (FTA_SSL) ... 41
5.8.2.1 FTA_SSL.3 TSF-initiated Termination ... 41
5.8.2.2 FTA_SSL.4 User-initiated Termination ... 41
5.8.3 TOE access banners (FTA_TAB) ... 42
5.8.3.1 FTA_TAB.1 Default TOE Access Banners ... 42
5.9 Trusted path/channels (FTP) ... 42
5.9.1 Trusted Channel (FTP_ITC) ... 42
5.9.1.1 FTP_ITC.1 Inter-TSF trusted channel (Refined) ... 42
5.9.2 Trusted Path (FTP_TRP) ... 43
5.9.2.1 FTP_TRP.1 Trusted Path (Refinement) ... 43

6. Security Assurance Requirements ... 49
6.1 ASE: Security Target ... 49
6.2 ADV: Development ... 50
6.2.1 Basic Functional Specification (ADV_FSP.1) ... 50
6.3 AGD: Guidance Documentation ... 50
6.3.1 Operational User Guidance (AGD_OPE.1) ... 51

6.3.2 Preparative Procedures (AGD_PRE.1) ... 51
6.4 Class ALC: Life-cycle Support ... 51
6.4.1 Labelling of the TOE (ALC_CMC.1) ... 51
6.4.2 TOE CM Coverage (ALC_CMS.1) ... 51
6.5 Class ATE: Tests ... 51
6.5.1 Independent Testing – Conformance (ATE_IND.1) ... 52
6.6 Class AVA: Vulnerability Assessment ... 52
6.6.1 Vulnerability Survey (AVA_VAN.1) ... 52
A. Optional Requirements ... 53
A.1 Audit Events for Optional SFRs ... 53
A.2 Security Audit (FAU) ... 53
A.2.1 Security audit event storage (Extended – FAU_STG_EXT) ... 53
A.2.1.1 FAU_ STG_EXT.2 Counting lost audit data ... 53
A.3 Security Management (FMT) ... 54
A.3.1 Management of functions in TSF (FMT_MOF) ... 54
A.3.1.1 FMT_MOF.1 Management of security functions behaviour ... 54
A.3.2 Management of TSF data (FMT_MTD) ... 55
A.3.2.1 FMT_MTD.1/AdminAct Management of TSF data ... 55
A.4 Protection of the TSF (FPT) ... 55
A.4.1 Fail Secure (FPT_FLS) ... 55
A.4.1.1 FPT_FLS.1/LocSpace Failure with preservation of secure state ... 55
B. Selection-Based Requirements ... 56
B.1 Audit Events for Selection-Based SFRs ... 56
B.2 Cryptographic Support (FCS) ... 57
B.2.1 Cryptographic Protocols (Extended – FCS_HTTPS_EXT, FCS_ IPSEC_EXT, FCS_SSHC_EXT, FCS_SSHS_EXT, FCS_TLSC_EXT, FCS_TLSS_EXT) ... 57
B.2.1.1 FCS_HTTPS_EXT.1 HTTPS Protocol ... 57
B.2.1.2 FCS_IPSEC_EXT.1 IPsec Protocol ... 57
B.2.1.3 FCS_SSHC_EXT.1 SSH Client Protocol ... 61
B.2.1.4 FCS_SSHS_EXT.1 SSH Server Protocol ... 63
B.2.1.5 FCS_TLSC_EXT.1 TLS Client Protocol ... 64
6.6.1.1 FCS_TLSS_EXT.1 TLS Server Protocol ... 66
B.3 Protection of the TSF (FPT) ... 68
B.3.1 TSF self test (Extended) ... 68
B.3.1.1 FPT_TST_EXT.2 Self tests based on certificates ... 68
B.3.2 Trusted Update (FPT_TUD_EXT) ... 69
B.3.2.1 FPT_TUD_EXT.2 Trusted Update based on certificates ... 69
C. Extended Component Definitions ... 70
C.1 Security Audit (FAU) ... 70
C.1.1 Security audit event storage (FAU_STG_EXT) ... 70
C.2 Cryptographic Support (FCS) ... 71
C.2.1 Random Bit Generation (FCS_RBG_EXT) ... 71
C.2.1.1 FCS_RBG_EXT.1 Random Bit Generation ... 71
C.2.2 Cryptographic Protocols (Extended – FCS_HTTPS_EXT, FCS_ IPSEC_EXT, FCS_SSHC_EXT, FCS_SSHS_EXT, FCS_TLSC_EXT, FCS_TLSS_EXT) ... 72
C.2.2.1 FCS_HTTPS_EXT.1 HTTPS Protocol ... 72
C.2.2.2 FCS_IPSEC_EXT.1 IPsec Protocol ... 73
C.2.2.3 FCS_SSHC_EXT.1 SSH Client ... 77
C.2.2.4 FCS_SSHS_EXT.1 SSH Server Protocol ... 79
C.2.2.5 FCS_TLSC_EXT.1 TLS Client Protocol ... 81
C.2.2.6 FCS_TLSS_EXT.1 TLS Server Protocol ... 83
C.3 Identification and Authentication (FIA) ... 87
C.3.1 Authentication using X.509 certificates (Extended – FIA_X509_EXT) ... 89
C.3.1.1 FIA_X509_EXT.1 X.509 Certificate Validation ... 90
C.3.1.2 FIA_X509_EXT.2 X509 Certificate Authentication ... 91
C.3.1.3 FIA_X509_EXT.3 X.509 Certificate Requests ... 92
C.4 Protection of the TSF (FPT) ... 92
C.4.1 TSF self test ... 94

C.4.1.1 TSF Testing (Extended) ... 94
C.4.1.2 FPT_TST_EXT.2 Self tests based on certificates .. **Error! Bookmark not defined.**
C.4.2 Trusted Update (FPT_TUD_EXT) ... 95
C.4.2.1 FPT_TUD_EXT.1 Trusted Update ... 96
C.4.2.2 FPT_TUD_EXT.2 Trusted Update based on certificates ... 97
D. Entropy Documentation And Assessment ... 99
D.1 Design Description ... 99
D.2 Entropy Justification ... 99
D.3 Operating Conditions ... 99
D.4 Health Testing ... 100
E. Glossary ... 101
F. Acronyms ... 102

Figures / Tables

Table 1: Security Functional Requirements and Auditable Events 25
Table 2: Security Assurance Requirements 49
Table 3: TOE Optional SFRs and Auditable Events........ 53
Table 4: Selection-Dependent SFRs and Auditable Events........ 56

1. PP Introduction

1.1 PP Reference Identification

PP Reference: collaborative Protection Profile for Stateful Traffic Filter Firewalls

PP Version: 0.1

PP Date: 05-Sep-2014

1.2 TOE Overview

This collaborative Protection Profile (cPP) defines requirements for the evaluation of Stateful Traffic Filter Firewalls. Such products are generally boundary protection devices, such as dedicated firewalls, routers, or perhaps even switches designed to control the flow of information between attached networks. While in some cases, firewalls implementing security features serve to segregate two distinct networks – a trusted or protected enclave and an untrusted internal or external network such as the Internet – that is only one of many possible applications. It is common for firewalls to have multiple physical network connections enabling a wide range of possible configurations and network information flow policies.

1.3 TOE Usage

This PP specifically addresses firewalls that perform network layer 3 and 4 stateful traffic filtering. A stateful traffic filter firewall is a device composed of hardware and software that is connected to two or more distinct networks and has an infrastructure role in the overall enterprise network.

Stateful traffic filtering is the idea that the firewall would keep track of the state of each connection through it and have the ability to drop packets that do not appear to belong to a valid flow. Information such as the TCP sequence number, ACKs, IP options are also kept by storing the metrics in dynamic state tables. Other considerations in the decision to accept, drop, or log packets are source and destination IP addresses and ports, or when the source or destination addresses are inconsistent with the configured interfaces.

Future drafts of this PP are envisioned, which will include optional functionality (e.g., transparent mode). Future Firewall PPs will be used to specify sets of additional functionality (e.g., Application Filtering). In the context of this PP, additional features such as these are simply ignored for the purpose of evaluation except where they may have some effect of the security requirements defined herein. While many devices that will be evaluated against this PP have the capability to perform NAT or PAT, there are no requirements that specify this capability.

2. CC Conformance

As defined by the references [CC1], [CC2] and [CC3], this cPP:

- conforms to the requirements of Common Criteria v3.1, Revision 4
- is Part 2 extended, Part 3 conformant
- does not claim conformance to any other PP.

The methodology applied for the cPP evaluation is defined in [CEM]. This cPP satisfies the following Assurance Families: APE_CCL.1, APE_ECD.1, APE_INT.1, APE_OBJ.1, APE_REQ.1 and APE_SPD.1.

In order to be conformant to this cPP, a TOE must demonstrate Exact Compliance. Exact Compliance, as a subset of Strict Compliance as defined by the CC, is defined as the ST containing all of the requirements in section 5 of the this cPP, and potentially requirements from Appendix A or Appendix B of this cPP. While iteration is allowed, no additional requirements (from the CC parts 2 or 3) are allowed to be included in the ST. Further, no requirements in section 5 of this cPP are allowed to be omitted.

3. Security Problem Definition

A stateful traffic filter firewall (defined to be a device that filters layers 3 and 4 (IP and TCP/UDP) network traffic optimized through the use of stateful packet inspection) is intended to provide a minimal, baseline set of requirements that are targeted at mitigating well defined and described threats.

It has the ability to match packets to a known active (and allowed) connection to permit them and drop others. The firewall often serves as a boundary device between two separate network security domains, and, as such, must provide a minimal set of common security functionality. These functional requirements define authorized communication with the firewall, audit capabilities, user access, update processes, and self-test procedures for critical components.

3.1 Threats

The threats for the stateful traffic filter firewall are grouped according to functional areas of the device in the sections below.

3.1.1 Communications with the Firewall

A firewall communicates with other network devices and other network entities. The endpoints of this communication can be geographically and logically distant and may pass through a variety of other systems. The intermediate systems may be untrusted providing an opportunity for unauthorized communication with the firewall or for authorized communication to be compromised. The security functionality of the firewall must be able to protect any critical network traffic (administration traffic, authentication traffic, audit traffic, etc.). The communication with the firewall falls into two categories: authorized communication and unauthorized communication.

Authorized communication includes normal network traffic allowable by policy destined to and originating from the firewall as it was designed and intended. This includes critical network traffic, such as firewall administration and communication with an authentication or audit logging server, which requires a secure channel to protect the communication. The security functionality of the firewall includes the capability to ensure that only authorized communications are allowed and the capability to provide a secure channel for critical network traffic. Any other communication is considered unauthorized communication.

The primary threats to firewall communications addressed in this cPP focus on an external, unauthorized entity attempting to access, modify, or otherwise disclose the critical network traffic. A poor choice of cryptographic algorithms or the use of non-standardized tunneling protocols along with weak administrator credentials, such as an easily guessable password or use of a default password, will allow a threat agent unauthorized access to the firewall. Weak or no cryptography provides little to no protection of the traffic allowing a threat agent to read, manipulate and/or control the critical data with little effort. Non-standardized tunneling protocols not only limit the interoperability of the firewall but lack the assurance and confidence standardization provides through peer review.

3.1.1.1 T.UNAUTHORIZED_ADMINISTRATOR_ACCESS

Threat agents may attempt to gain administrator access to the firewall by nefarious means such as masquerading as an administrator to the firewall, masquerading as the firewall to an administrator, replaying an administrative session (in its entirety, or selected portions), or performing man-in-the-middle attacks, which would provide access to the administrative session, or sessions between the firewall and a network device. Successfully gaining administrator access allows malicious actions that compromise the security functionality of the firewall and the network on which it resides.

3.1.1.2 T.WEAK_CRYPTOGRAPHY

Threat agents may exploit weak cryptographic algorithms or perform a cryptographic exhaust against the key space. Poorly chosen encryption algorithms, modes, and key sizes will allow attackers to compromise the algorithms, or brute force exhaust the key space and give them unauthorized access allowing them to read, manipulate and/or control the traffic with minimal effort.

3.1.1.3 T.UNTRUSTED_COMMUNICATION_CHANNELS

Threat agents may attempt to target firewalls that do not use standardized secure tunneling protocols to protect the critical network traffic. Attackers may take advantage of poorly designed protocols or poor key management to successfully perform man-in-the-middle attacks, replay attacks, etc. Successful attacks will result in loss of confidentiality and integrity of the critical network traffic, and potentially could lead to a compromise of the firewall itself.

3.1.1.4 T.WEAK_AUTHENTICATION_ENDPOINTS

Threat agents may take advantage of secure protocols that use weak methods to authenticate the endpoints – e.g., shared password that is guessable or transported as plaintext. The consequences are the same as a poorly designed protocol, the attacker could masquerade as the administrator or another device, and the attacker could insert themselves into the network stream and perform a man-in-the-middle attack. The result is the critical network traffic is exposed and there could be a loss of confidentiality and integrity, and potentially the firewall itself could be compromised.

3.1.2 Valid Updates

Updating firewall software and firmware is necessary to ensure that the security functionality of the firewall is maintained. The source and content of an update to be applied must be validated by cryptographic means; otherwise, an invalid source can write their own firmware or software updates that circumvents the security functionality of the firewall. Methods of validating the source and content of a software or firmware update by cryptographic means typically involve cryptographic signature schemes where hashes of the updates are digitally signed.

Unpatched versions of software or firmware leave the firewall susceptible to threat agents attempting to circumvent the security functionality using known vulnerabilities. Non-validated updates or updates validated using non-secure or weak cryptography leave the updated software or firmware vulnerable to threat agents attempting to modify the software or firmware to their advantage.

3.1.2.1 T.UPDATE_COMPROMISE

Threat agents may attempt to provide a compromised update of the software or firmware which undermines the security functionality of the device. Non-validated updates or updates validated using non-secure or weak cryptography leave the update firmware vulnerable to surreptitious alteration.

3.1.3 Audited Activity

Auditing of firewall activities is a valuable tool for administrators to monitor the status of the device. It provides the means for administrator accountability, security functionality activity reporting, reconstruction of events, and problem analysis. Processing performed in response to device activities may give indications of a failure or compromise of the security functionality. When indications of activity that impact the security functionality are not generated and monitored it is possible for such activities to occur without administrator awareness. Further, if records are not generated or retained, reconstruction of the network and the ability to understand the extent of any compromise could be negatively affected. Additional concerns are the protection of the audit data that is recorded from alteration or unauthorized deletion. This could occur within the TOE, or while the audit data is in transit to an external storage device.

Note this cPP requires that the firewall generate the audit data and have the capability to send the audit data to a trusted network entity (e.g., a *syslog* server).

3.1.3.1 T.UNDETECTED_ACTIVITY

Threat agents may attempt to access, change, and/or modify the security functionality of the firewall without administrator awareness. This could result in the attacker finding an avenue (e.g., misconfiguration, flaw in the product) to compromise the device and the administrator would have no knowledge that the device has been compromised.

3.1.4 Administrator and Firewall Credentials and Data

A firewall contains data and credentials which must be securely stored and must appropriately restrict access to authorized entities. Examples include the firewall firmware, software, configuration, authentication credentials for secure channels, and administrator credentials. Firewall and administrator keys, key material, and authentication credentials need to be protected from unauthorized disclosure and modification. Furthermore, the security functionality of the firewall needs to require default authentication credentials, such as administrator passwords, be changed.

Lack of secure storage and improper handling of credentials and data, such as unencrypted credentials inside configuration files or access to secure channel session keys, can allow an attacker to not only gain access to the firewall, but also compromise the security of the network through seemingly authorized modifications to configuration or though man-in-the-middle attacks. These attacks allow an unauthorized entity to gain access and perform administrative functions using the authorized administrator's credentials and to intercept all traffic as an authorized endpoint. This results in difficulty in detection of security compromise and in reconstruction of the network, potentially allowing continued unauthorized access to administrator and firewall data.

3.1.4.1 T.SECURITY_FUNCTIONALITY_COMPROMISE

Threat agents may compromise credentials and firewall data enabling continued access to the firewall and its critical data. The compromise of credentials include replacing existing credentials with an attacker's credentials, modifying existing credentials, or obtaining the administrator or firewall credentials for use by the attacker.

3.1.4.2 T.PASSWORD_CRACKING

Threat agents may be able to take advantage of weak administrative passwords to gain privileged access to the firewall. Having privileged access to the firewall provides the attacker unfettered access to the network traffic, and may allow them to take advantage of any trust relationships with other network devices.

3.1.5 Firewall Component Failure

Security mechanisms of the firewall generally build up from roots of trust to more complex sets of mechanisms. Failures could result in a compromise to the security functionality of the firewall. A firewall self-testing its security critical components at both start-up and during run-time ensures the reliability of the firewall's security functionality.

3.1.5.1 T.SECURITY_FUNCTIONALITY_FAILURE

A component of the firewall may fail during start-up or during operations causing a compromise or failure in the security functionality of the firewall, leaving the firewall susceptible to attackers. Avenues of attack could be opened such as the cryptographic functions no longer properly working, including random number generation, allowing an attacker to connect to the firewall.

3.2 Unauthorized Disclosure of Information

Devices on a protected network may be exposed to threats presented by devices located outside the protected network, which may attempt to conduct unauthorized activities. If known malicious external devices are able to communicate with devices on the protected network, or if devices on the protected network can establish communications with those external devices, then those internal devices may be susceptible to the unauthorized disclosure of information.

From an infiltration perspective, Stateful Traffic Filter Firewalls serve to limit access to only specific *destination* network addresses and ports within a protected network. With these limits, general network port scanning can be prevented from reaching protected networks or machines, and access to information on a protected network can be limited to that obtainable from specifically configured ports on identified network nodes (e.g., web pages from a designated corporate web server). Additionally, access can be limited to only specific *source* addresses and ports so that specific networks or network nodes can be blocked from accessing a protected network thereby further limiting the potential disclosure of information.

From an exfiltration perspective, Stateful Traffic Filter Firewalls serve to limit how network nodes operating on a protected network can connect to and communicate with other networks limiting how and where they can disseminate information. Specific external networks can be blocked altogether or egress could be limited to specific addresses and/or ports. Alternately, egress options available to network nodes on a protected network can be carefully managed in order to, for example, ensure that outgoing connections are routed through authorized proxies or filters to further mitigate inappropriate disclosure of data through extrusion.

3.2.1 T.NETWORK_DISCLOSURE

An attacker may attempt to “map” a subnet to determine the machines that reside on the network, and obtaining the IP addresses of machines, as well as the services (ports) those machines are offering. This information could be used to mount attacks to those machines via the services that are exported.

3.3 Inappropriate Access to Services

Devices located outside the protected network may seek to exercise services located on the protected network that are intended to only be accessed from inside the protected network. Devices located outside the protected network may, likewise, offer services that are inappropriate for access from within the protected network.

From an ingress perspective, Stateful Traffic Filter Firewalls can be configured so that only those network servers intended for external consumption are accessible and only via the intended ports. This serves to mitigate the potential for network entities outside a protected network to access network servers or services intended only for consumption or access inside a protected network.

From an egress perspective, Stateful Traffic Filter Firewalls can be configured so that only specific external services (e.g., based on destination port) can be accessed from within a protected network. For example, access to external mail services can be blocked to enforce corporate policies against accessing uncontrolled e-mail servers. Note that the effectiveness of a Stateful Traffic Filter Firewall is rather limited in this regard since external servers can offer their services on alternate ports – this is where an Application Filter Firewall offers more reliable protection, for example.

3.3.1 T. NETWORK_ACCESS

With knowledge of the services that are exported by machines on a subnet, an attacker may attempt to exploit those services by mounting attacks against those services.

3.4 Misuse of Services

Devices located outside a “protected” network, while permitted to access particular *public* services offered inside the protected network, may attempt to conduct inappropriate activities while communicating with those allowed public services. Certain services offered from within a protected network may also represent a risk when accessed from outside the protected network. It should be noted that the firewall simply enforces rules that are specified for a network interface. The notion of a protected or trusted network is an abstraction that is useful when constructing the ruleset.

From an ingress perspective, it is generally assumed that entities operating on external networks are not bound by the use policies for a given protected network. Nonetheless, Stateful Traffic Filter Firewalls can log policy violations that might indicate violation of publicized usage statements for publicly available services.

From an egress perspective, Stateful Traffic Filter Firewalls can be configured to help enforce and monitor protected network use policies. As explained in the other threats, a Stateful Traffic Filter Firewall can serve to limit dissemination of data, access to external servers, and even disruption of services – all of these could be related to the use policies of a protected network and as such are subject in some regards to enforcement. Additionally, Stateful Traffic Filter Firewalls can be configured to log network usages that cross between protected and external networks and as a result can serve to identify potential usage policy violations.

3.4.1 T.NETWORK_MISUSE

An attacker may attempt to use services that are exported by machines in a way that is unintended by a site's security policies. For example, an attacker might be able to use a service to "anonymize" the attacker's machine as they mount attacks against others.

3.5 Malicious Traffic

A stateful traffic filtering firewall also provides protections against malicious or malformed packets. It will protect against attacks like modification of connection state information and replay attacks. These attacks could cause the firewall, or the devices it protects, to grant unauthorized access or even create a Denial of Service.

3.5.1 T.MALICIOUS_TRAFFIC

An attacker may attempt to send malformed packets to a machine in hopes of causing the network stack or services listening on UDP/TCP ports of the target machine to crash,

3.6 Assumptions

This section describes the assumptions made in identification of the threats and security requirements for firewalls. The firewall is not expected to provide assurance in any of these areas, and as a result, requirements are not included to mitigate the threats associated.

3.6.1 A.PHYSICAL_PROTECTION

The firewall is assumed to be physically protected in its operational environment and not subject to physical attacks that compromise the security and/or interfere with the firewall's physical interconnections and correct operation. This protection is assumed to be sufficient to protect the firewall and the data it contains. As a result, the cPP will not include any requirements on physical tamper protection or other physical attack mitigations. The cPP will not expect the product to defend against physical access to the firewall that allows unauthorized entities to extract data, bypass other controls, or otherwise manipulate the firewall.

[OE.PHYSICAL]

3.6.2 A.LIMITED_FUNCTIONALITY

The firewall is assumed to provide networking and filtering functionality as its core function and not provide functionality/services that could be deemed as general purpose computing. For example the firewall should not provide computing platform for general purpose applications (unrelated to networking/filtering functionality).

[OE.NO_GENERAL_PURPOSE]

3.6.3 A.TRUSTED_ADMINSTRATOR

The authorized administrator(s) for the firewall are assumed to be trusted and to act in the best interest of security for the organization. This includes being appropriately trained, following policy, and adhering to administrator guidance. Administrators are trusted to ensure passwords/credentials have sufficient strength and entropy and to lack malicious intent when administering the firewall. The firewall is not expected to be capable of defending against a malicious administrator that actively works to bypass or compromise the security of the firewall.

[OE.TRUSTED_ADMIN]

3.6.4 A.REGULAR_UPDATES

The firewall firmware and software is assumed to be updated by an administrator on a regular basis in response to the release of product updates due to known vulnerabilities.

[OE.UPDATES]

3.6.5 A.ADMIN_CREDENTIALS_SECURE

The administrator’s credentials (private key) used to access the firewall are protected by the host platform on which they reside.

[OE.ADMIN_CREDENTIALS_SECURE]

3.7 Organizational Security Policy

An organizational security policy is a set of rules, practices, and procedures imposed by an organization to address its security needs. For the purposes of this cPP a single policy is described in the section below.

3.7.1 P.ACCESS_BANNER

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

[FTA_TAB.1]

4. Security Objectives

4.1 Security Objectives for the Operational Environment

The following subsections describe objectives for the Operational Environment.

4.1.1 OE.PHYSICAL

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

4.1.2 OE.NO_GENERAL_PURPOSE

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

4.1.3 OE.TRUSTED_ADMIN

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

4.1.4 OE.UPDATES

The TOE firmware and software is updated by an administrator on a regular basis in response to the release of product updates due to known vulnerabilities.

4.1.5 OE.ADMIN_CREDENTIALS_SECURE

The administrator's credentials (private key) used to access the TOE must be protected on any other platform on which they reside.

5. Security Functional Requirements

The individual security functional requirements are specified in the sections below.

The Evaluation Activities defined in [SD] describe actions that the evaluator will take in order to determine compliance of a particular TOE with the SFRs. The content of these Evaluation Activities will therefore provide more insight into deliverables required from TOE Developers.

5.1 Conventions

The conventions used in descriptions of the SFRs are as follows:

- Assignment: Indicated with *italicized* text;
- Refinement made by PP author: Indicated with **bold text** and ~~strikethroughs~~, if necessary;
- Selection: Indicated with <u>underlined text</u>;
- Assignment within a Selection: Indicated with <u>*italicized and underlined text*</u>;
- Iteration: Indicated by appending the iteration number in parenthesis, e.g., (1), (2), (3) and/or by adding a string starting with "/".

Extended SFRs are identified by having a label 'EXT' at the end of the SFR name.

5.2 Security Audit (FAU)

5.2.1 Security Audit Data generation (FAU_GEN)

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

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

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

5.2.1.1 FAU_GEN.1 Audit data generation

FAU_GEN.1	Audit Data Generation

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

a) Start-up and shut-down of the audit functions;

b) All auditable events for the <u>not specified</u> level of audit; and

c) *All administrative actions*;

d) *Specifically defined auditable events listed in Table 1*.

Application Note 1

The term 'administrative actions' comprises:

- *Administrative login and logout (name of user account shall be logged if individual user accounts are required for administrators).*
- *Configuration changes (in addition to the information that a change occurred it shall be logged what has been changed).*
- *Generating/import of, changing, or deleting of cryptographic keys (optional: key export) (in addition to the action itself a unique key name or key reference shall be logged).*
- *Changing passwords (name of related user account shall be logged).*
- *Starting and stopping services (if applicable)*
- *Other uses of privileges.*

The ST author replaces the cross-reference to the table of audit events with an appropriate cross-reference for the ST. This must also include the relevant parts of Table 3 and Table 4 for optional and selection-dependent SFRs included in the ST.

Application Note 2

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

The TSS should identify what information is logged to identify the relevant key for the administrative task of generating/import of, changing, or deleting of cryptographic keys.

Starting and stopping services refers to regular activities. In case of unforeseen events like the crash of the audit service, it might not be possible to generate or store audit data.

With respect to FAU_GEN.1.1 the term 'services' refers to e.g. audit service, SSH server, SNMP agent, NETCONF, routing protocol daemons, update service.

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

a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, *information specified in column three of Table 1*.

Application Note 3

The ST author replaces the cross-reference to the table of audit events with an appropriate cross-reference for the ST. This must also include the relevant parts of Table 3 and Table 4 for optional and selection-dependent SFRs included in the ST.

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None.	None.
FAU_GEN.2	None.	None.
FAU_STG_EXT.1	None.	None.
FCS_CKM.4	None.	None.
FCS_COP.1(1)	None.	None.
FCS_COP.1(2)	None.	None.
FCS_COP.1(3)	None.	None.
FCS_COP.1(4)	None.	None.
FCS_RBG_EXT.1	None.	None.
FDP_RIP.2	None.	None.
FFW_RUL_EXT.1	Application of rules configured with the 'log' operation	Source and destination addresses Source and destination ports Transport Layer Protocol TOE Interface
	Indication of packets dropped due to too much	TOE interface that is unable to process packets

	network traffic	Identifier of rule causing packet drop
FIA_PMG_EXT.1	None.	None.
FIA_UIA_EXT.1	All use of identification and authentication mechanism.	Provided user identity, origin of the attempt (e.g., IP address).
FIA_UAU_EXT.2	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UAU.7	None.	None.
FIA_X509_EXT.1	Failure to validate a certificate	Reason for failure
FIA_X509_EXT.2	None	None
FMT_MOF.1	None.**	None.**
FMT_MTD.1	None.**	None.**
FMT_SMF.1	None.	None.
FMT_SMR.2	None.	None.
FPT_APW_EXT.1	None.	None.
FPT_ITT.1	None.	None.
FPT_SKP_EXT.1	None.	None.
FPT_STM.1	Changes to time.	The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address).
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure)	No additional information.
FPT_TUD_EXT.2	None.	None.
FPT_TST_EXT.1	None.	None.
FPT_TST_EXT.2	None.	None.

FTA_SSL_EXT.1	Any attempts at unlocking of an interactive session.	None.
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	None.
FTA_SSL.4	The termination of an interactive session.	None.
FTA_TAB.1	None.	None.
FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
FTP_TRP.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the claimed user identity.

Table 1: Security Functional Requirements and Auditable Events

Application Note 4

Additional audit events will apply to the TOE depending on the optional and selection-based requirements adopted from Appendix A and Appendix B. The ST author must therefore include the relevant additional events specified in the tables in Table 3 and Table 4.

In Table 1 (and the other tables of audit events in Appendix A and Appendix B):

> ***: 'None' in this case means that no events are logged in addition to the events that are logged for 'administrative actions' as defined in FAU_GEN.1.1.*

5.2.1.2 FAU_GEN.2 User identity association

FAU_GEN.2	**User identity association**

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

5.2.2 Security audit event storage (Extended – FAU_STG_EXT)

5.2.2.1 FAU_ STG_EXT.1 External Audit Trail Storage

FAU_STG_EXT.1	Protected Audit Trail Storage

FAU_STG_EXT.1.1 The TSF shall be able to transmit the generated audit data to an external IT entity using a trusted channel implementing the [selection: *IPsec, SSH, TLS, TLS/HTTPS*] protocol.

Application Note 5

For selecting the option of transmission of generated audit data to an external IT entity the TOE relies on a non-TOE audit server for storage and review of audit records. The storage of these audit records and the ability to allow the administrator to review these audit records is provided by the operational environment in that case.

In the second selection, the ST author chooses the means by which this connection is protected. The ST author also has to ensure that the supporting protocol requirement matching the selection is included in the ST.

FAU_STG_EXT.1.2 The TSF shall be able to store generated audit date on the TOE itself.

Application Note 6

The local space to store audit date is limited. The TSF shall generate a warning to inform the user before the local space to store audit data is used up and/or the TOE will lose audit data due to insufficient local space.

FAU_STG_EXT.1.3 The TSF shall [selection: *drop new audit data, overwrite previous audit records according to the following rule: [assignment: rule for overwriting previous audit records], [assignment: other action]*] when the local storage space for audit data is full.

Application Note 7

The external log server might be used as alternative storage space in case the local storage space is full. The 'other action' could in this case be defined as 'send the new audit date to an external IT entity'.

5.3 Cryptographic Support (FCS)

5.3.1 Cryptographic Key Management (FCS_CKM)

5.3.1.1 FCS_CKM.1 Cryptographic Key Generation

FCS_CKM.1	Cryptographic Key Generation

FCS_CKM.1.1 The TSF shall generate **asymmetric** cryptographic keys in accordance with a specified cryptographic key generation algorithm: [selection:

- ***RSA schemes*** *using cryptographic key sizes of* ***2048-bit or greater*** *that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3;*

- ***ECC schemes*** *using* ***"NIST curves" P-256, P-384 and [selection: P-521, no other curves]*** *that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4;*

- ***FFC schemes*** *using cryptographic key sizes of* ***2048-bit or greater*** *that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.1*

].

Application Note 8

The ST author shall select all key generation schemes used for key establishment and device authentication. When key generation is used for key establishment, the schemes in FCS_CKM.2.1 and selected cryptographic protocols must match the selection. When key generation is used for device authentication, the public key is expected to be associated with an X.509v3 certificate.

If the TOE acts as a receiver in the RSA key establishment scheme, the TOE does not need to implement RSA key generation.

5.3.1.2 FCS_CKM.2 Cryptographic Key Establishment

FCS_CKM.2	Cryptographic Key Establishment

FCS_CKM.2.1 The TSF shall perform cryptographic key establishment in accordance with a specified cryptographic key establishment method: [selection:

- *RSA-based key establishment schemes that meets the following: NIST Special Publication 800-56B, "Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography";*

- *Elliptic curve-based key establishment schemes that meets the following: NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography";*

- *Finite field-based key establishment schemes that meets the following: NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"*

].

Application Note 9

This is a refinement of the SFR FCS_CKM.2 to deal with key establishment rather than key distribution.

The ST author shall select all key establishment schemes used for the selected cryptographic protocols.

The RSA-based key establishment schemes are described in Section 9 of NIST SP 800-56B; however, Section 9 relies on implementation of other sections in SP 800-56B. If the TOE acts as a receiver in the RSA key establishment scheme, the TOE does not need to implement RSA key generation.

The elliptic curves used for the key establishment scheme shall correlate with the curves specified in FCS_CKM.1.1.

The domain parameters used for the finite field-based key establishment scheme are specified by the key generation according to FCS_CKM.1.1.

5.3.1.3 FCS_CKM.4 Cryptographic Key Destruction

FCS_CKM.4	**Cryptographic Key Destruction**

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [selection:

- *For volatile memory, the destruction shall be executed by a single direct overwrite [selection: consisting of a pseudo-random pattern using the TSF's RBG, consisting of zeroes] followed by a read-verify.*
    - *If the read-verification of the overwritten data fails, the process shall be repeated again.*
- *For non-volatile EEPROM, the destruction shall be executed by a single, direct overwrite consisting of a pseudo random pattern using the TSF's RBG (as specified in FCS_RBG_EXT.1), followed by a read-verify.*
    - *If the read-verification of the overwritten data fails, the process shall be repeated again.*
- *For non-volatile flash memory, the destruction shall be executed by [selection: a single, direct overwrite consisting of zeroes, a block erase] followed by a read-verify.*
    - *If the read-verification of the overwritten data fails, the process shall be repeated again.*
- *For non-volatile memory other than EEPROM and flash, the destruction shall be executed by overwriting three or more times with a random pattern that is changed before each write.*

]

that meets the following: *NIST SP 800-88*.

5.3.2 Cryptographic Operation (FCS_COP)

5.3.2.1 FCS_COP.1 Cryptographic Operation

FCS_COP.1(1)	**Cryptographic Operation (AES Data Encryption/ Decryption)**

FCS_COP.1.1(1) The TSF shall perform *encryption/decryption* in accordance with a specified cryptographic algorithm *AES used in [selection: CBC, GCM] mode* and cryptographic key sizes *[selection: 128 bits, 192 bits, 256 bits]* that meet the following: *AES as specified in ISO 18033-3, [selection: CBC as specified in ISO 10116, GCM as specified in ISO 19772]*.

Application Note 10

For the first selection of FCS_COP.1.1(1), the ST author should choose the mode or modes in which AES operates. For the second selection, the ST author should choose the key sizes that are supported by this functionality. The modes and key sizes selected here correspond to the cipher suite selections made in the trusted channel requirements.

FCS_COP.1(2)	**Cryptographic Operation (Signature Verification)**

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

- *RSA Digital Signature Algorithm and cryptographic key sizes* ***(modulus)*** *[assignment: 2048 bits or greater],*
- *Elliptic Curve Digital Signature Algorithm and cryptographic key sizes [assignment: 256 bits or greater]*

]

that meets the following: [selection:

- *For RSA schemes: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS2v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3,*
- *For ECDSA schemes: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 6 and Appendix D, Implementing "NIST curves" P-256, P-384, and [selection: P-521, no other curves]; ISO/IEC 14888-3, Section 6.4*

].

Application Note 11

The ST Author should choose the algorithm implemented to perform digital signatures. For the algorithm(s) chosen, the ST author should make the appropriate assignments/selections to specify the parameters that are implemented for that algorithm.

FCS_COP.1(3)	**Cryptographic Operation (Hash Algorithm)**

FCS_COP.1.1(3) The TSF shall perform *cryptographic hashing services* in accordance with a specified cryptographic algorithm [*selection: SHA-1, SHA-256, SHA-384, SHA-512, no other algorithms*] ~~**and cryptographic key sizes [assignment: *cryptographic key sizes*]**~~ that meet the following: *ISO/IEC 10118-3:2004*.

Application Note 12

Vendors are strongly encouraged to implement updated protocols that support the SHA-2 family; until updated protocols are supported, this PP allows support for SHA-1 implementations in compliance with SP 800-131A.

The hash selection should be consistent with the overall strength of the algorithm used for FCS_COP.1(1) and FCS_COP.1(2) (for example, SHA 256 for 128-bit keys). The selection of the standard is made based on the algorithms selected.

FCS_COP.1(4)	**Cryptographic Operation (Keyed Hash Algorithm)**

FCS_COP.1.1(4) The TSF shall perform *keyed-hash message authentication* in accordance with a specified cryptographic algorithm *[selection: HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512, no other algorithms]* and cryptographic key sizes [assignment: <u>key size (in bits) used in HMAC</u>] and **message digest sizes** *160 and [selection: 256, 384, 512, no other] bits* that meet the following: *ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2"*.

Application Note 13

The key size [k] in the assignment falls into a range between L1 and L2 (defined in ISO/IEC 10118 for the appropriate hash function. For example, for SHA-256, L1=512, L2=256, where L2<=k<=L1.

5.3.3 Random Bit Generation (Extended – FCS_RBG_EXT)

5.3.3.1 FCS_RBG_EXT.1 Random Bit Generation

FCS_RBG_EXT.1	**Random Bit Generation**

FCS_RBG_EXT.1.1 The TSF shall perform all deterministic random bit generation services in accordance with ISO/IEC 18031:2011 using [selection: *Hash_DRBG (any), HMAC_DRBG (any), CTR_DRBG (AES)*].

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded by an entropy source that accumulates entropy from [selection: *a software-based noise source*, *a hardware-based noise source*] with a minimum of [selection: *128 bits, 192 bits*, *256 bits*] of entropy at least equal to the greatest security strength according to ISO/IEC 18031:2011 Table C.1 "Security Strength Table for Hash Functions", of the keys and hashes that it will generate.

Application Note 14

ISO/IEC 18031:2011 contains three different methods of generating random numbers; each of these, in turn, depends on underlying cryptographic primitives (hash functions/ciphers). The ST author will select the function used, and include the specific underlying cryptographic primitives used in the requirement. While any of the identified hash functions (SHA-1, SHA-224, SHA-256, SHA-384, SHA-512) are allowed for Hash_DRBG or HMAC_DRBG, only AES-based implementations for CTR_DRBG are allowed.

If the key length for the AES implementation used here is different than that used to encrypt the user data, then FCS_COP.1 may have to be adjusted or iterated to reflect the different key length. For the selection in FCS_RBG_EXT.1.2, the ST author selects the minimum number of bits of entropy that is used to seed the RBG.

5.4 User Data Protection (FDP)

5.4.1 Residual information protection (FDP_RIP)

5.4.1.1 FDP_RIP.2 Full Residual Information Protection

FDP_RIP.2	Full Residual Information Protection

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

Application Note 15

"Resources" in the context of this requirement are network packets being sent through (as opposed to "to", as is the case when a security administrator connects to the TOE) the TOE. The concern is that once a network packet is sent, the buffer or memory area used by the packet still contains data from that packet, and that if that buffer is re-used, those data might remain and make their way into a new packet.

5.5 Identification and Authentication (FIA)

5.5.1 Password Management (Extended – FIA_PMG_EXT)

5.5.1.1 FIA_PMG_EXT.1 Password Management

FIA_PMG_EXT.1	Password Management

FIA_PMG_EXT.1.1 The TSF shall provide the following password management capabilities for administrative passwords:

1. *Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters:* [selection: *"!", "@", "#", "$", "%", "^", "&", "*", "(", ")", [assignment: other characters]];*
2. *Minimum password length shall settable by the Security Administrator, and support passwords of 15 characters or greater.*

Application Note 16

The ST author selects the special characters that are supported by TOE; they may optionally list additional special characters supported using the assignment. "Administrative passwords" refers to passwords used by administrators at the local console or over protocols that support passwords, such as SSH and HTTPS.

5.5.2 User Identification and Authentication (Extended – FIA_UIA_EXT)

5.5.2.1 FIA_UIA_EXT.1 User Identification and Authentication

FIA_UIA_EXT.1	User Identification and Authentication

FIA_UIA_EXT.1.1 The TSF shall allow the following actions prior to requiring the non-TOE entity to initiate the identification and authentication process:

- Display the warning banner in accordance with FTA_TAB.1;
- [selection: *no other actions, [assignment: list of services, actions performed by the TSF in response to non-TOE requests.]*]

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

Application Note 17

This requirement applies to users (administrators and external IT entities) of services available from the TOE directly, and not services available by connecting through the TOE. While it should be the case that few or no services are available to external entities prior to identification and authentication, if there are some available (perhaps ICMP echo) these *should be listed in the assignment statement; otherwise "no other actions" should be selected.*

Authentication can be password-based through the local console or through a protocol that supports passwords (such as SSH), or be certificate based (SSH, TLS).

For communications with external IT entities (e.g., an audit server or NTP server, for instance), such connections must be performed in accordance with FTP_ITC.1, whose protocols perform identification and authentication. This means that such communications (e.g., establishing the IPsec connection to the authentication server) would not have to be specified in the assignment, since establishing the connection "counts" as initiating the identification and authentication process.

5.5.3 User authentication (FIA_UAU) (Extended – FIA_UAU_EXT)

5.5.3.1 FIA_UAU_EXT.2 Password-based Authentication Mechanism

FIA_UAU_EXT.2	Password-based Authentication Mechanism

FIA_UAU_EXT.2.1 The TSF shall provide a local password-based authentication mechanism, [selection: *[assignment: other authentication mechanism(s)]*, *none*] to perform administrative user authentication.

5.5.3.2 FIA_UAU.7 Protected Authentication Feedback

FIA_UAU.7	Protected Authentication Feedback

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

Application Note 18

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

5.5.4 Authentication using X.509 certificates (Extended – FIA_X509_EXT)

5.5.4.1 FIA_X509_EXT.1 X.509 Certificate Validation

FIA_X509_EXT.1	X.509 Certificate Validation

FIA_X509_EXT.1.1 The TSF shall validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certificate path validation.

- The certificate path must terminate with a trusted certificate.
- The TSF shall validate a certificate path by ensuring the presence of the basicConstraints extension and that the CA flag is set to TRUE for all CA certificates.
- The TSF shall validate the revocation status of the certificate using [selection: *the Online Certificate Status Protocol (OCSP) as specified in RFC 2560, a Certificate Revocation List (CRL) as specified in RFC 5759*].
- The TSF shall validate the extendedKeyUsage field according to the following rules:
    - *Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.*
    - *Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.*
    - *Client certificates presented for TLS shall have 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.*

Application Note 19

FIA_X509_EXT.1.1 lists the rules for validating certificates. The ST author shall select whether revocation status is verified using OCSP or CRLs. FIA_X509_EXT.2 requires that certificates are used for IPsec; this use requires that the extendedKeyUsage rules are verified. Certificates may optionally be used for SSH, TLS and HTTPS and, if implemented, must be validated to contain the corresponding extendedKeyUsage.

Regardless of the selection of TSF or TOE platform, the validation is expected to end in a trusted root CA certificate in a root store managed by the platform.

FIA_X509_EXT.1.2 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 20

This requirement applies to certificates that are used and processed by the TSF and restricts the certificates that may be added as trusted CA certificates.

5.5.4.2 FIA_X509_EXT.2 X.509 Certificate Authentication

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 [selection*: IPsec, TLS, HTTPS, SSH*], and [selection: *code signing for system software updates, code signing for integrity verification, [assignment: other uses], no additional uses*].

Application Note 21

The ST author's selection shall match the selection of FTP_ITC.1.1 Certificates may optionally be used for trusted updates of system software (FPT_TUD_EXT.1) and for integrity verification (FPT_TST_EXT.2).

FIA_X509_EXT.2.2 When the TSF cannot establish a connection to determine the validity of a certificate, the TSF shall [selection: *allow the administrator to choose whether to accept the certificate in these cases, accept the certificate, not accept the certificate*].

Application Note 22

Often a connection must be established to check the revocation status of a certificate - either to download a CRL or to perform a lookup using OCSP. The selection is used to describe the behavior in the event that such a connection cannot be established (for example, due to a network error). If the TOE has determined the certificate valid according to all other rules in FIA_X509_EXT.1, the behavior indicated in the selection shall determine the validity. The TOE must not accept the certificate if it fails any of the other validation rules in FIA_X509_EXT.1. If the administrator-configured option is selected by the ST Author, the ST Author must also select the corresponding function in FMT_SMF.1.

5.5.4.3 FIA_X509_EXT.3 X.509 Certificate Requests

FIA_X509_EXT.3.1 The TSF shall generate a Certificate Request Message as specified by RFC 2986 and be able to provide the following information in the request: public key and [selection: *device-specific information, Common Name, Organization, Organizational Unit, Country*].

Application Note 23

The public key is the public key portion of the public-private key pair generated by the TOE as specified in FCS_CKM.1(1).

FIA_X509_EXT.3.2 The TSF shall validate the chain of certificates from the Root CA upon receiving the CA Certificate Response.

5.6 Security Management (FMT)

5.6.1 Management of functions in TSF (FMT_MOF)

5.6.1.1 FMT_MOF.1(1)/TrustedUpdate Management of TSF Data

FMT_MOF.1(1)/TrustedUpdate	Management of TSF Data

FMT_MOF.1.1(1)/TrustedUpdate The TSF shall restrict the ability to <u>enable</u> of the functions *perform manual update* to *Security Administrators*.

Application Note 24

FMT_MOF.1(1)/TrustedUpdate restricts the initiation of manual updates to Security Administrators.

5.6.2 Management of TSF Data (FMT_MTD)

5.6.2.1 FMT_MTD.1 Management of TSF Data

FMT_MTD.1	Management of TSF Data

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

Application Note 25

The word "manage" includes but is not limited to create, initialize, view, change default, modify, delete, clear, and append.

5.6.3 Specification of Management Functions (FMT_SMF)

5.6.3.1 FMT_SMF.1 Specification of Management Functions

FMT_SMF.1	Specification of Management Functions

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions:

- Ability to administer the TOE locally and remotely;
- Ability to update the TOE, and to verify the updates using digital signature capability prior to installing those updates;
- Ability to configure firewall rules;
- [selection:
  - *Ability to configure the list of TOE-provided services available before an entity is identified and authenticated, as specified in FIA_UIA_EXT.1;*
  - *Ability to configure the cryptographic functionality;*
  - *No other capabilities.*]

Application Note 26

The TOE must provide functionality for both local and remote administration, as well as the capability for the administrator to verify that updates received came from a trusted source. They must be capable of performing this action using digital signatures. If the TOE offers the ability for the administrator to configure the services available prior to identification or authentication, or if any of the cryptographic functionality on the TOE can be configured, then the ST author makes the appropriate choice or choices in the second selection, otherwise select "No other capabilities."

5.6.4 Security management roles (FMT_SMR)

5.6.4.1 FMT_SMR.2 Restrictions on security roles

FMT_SMR.2	Restrictions on Security Roles

FMT_SMR.2.1 The TSF shall maintain the roles:

- Security Administrator.

FMT_SMR.2.2 The TSF shall be able to associate users with roles.

FMT_SMR.2.3 The TSF shall ensure that the conditions

- Security Administrator role shall be able to administer the TOE locally;
- Security Administrator role shall be able to administer the TOE remotely;

are satisfied.

Application Note 27

FMT_SMR.2.2 requires that user accounts be associated with only one role. However, note that multiple users may have the same role, and the TOE is not required to restrict roles to a single person.

FMT_SMR.2.3 requires that a Security Administrator be able to administer the TOE through the local console and through a remote mechanism (IPsec, SSH, TLS, TLS/HTTPS). For multiple component TOEs, only the TOE components providing the management control and configuration of the other TOE components require a local administration interface.

5.7 Protection of the TSF (FPT)

5.7.1 Internal TOE TSF data transfer (FPT_ITT)

5.7.1.1 FPT_ITT.1 Basic Internal TSF Data Transfer Protection (Refinement)

FPT_ITT.1	Basic Internal TSF Data Transfer Protection

FPT_ITT.1.1 The TSF shall protect TSF data from disclosure and **detect its** modification when it is transmitted between separate parts of the TOE **through the use of [selection: choose at least one of: IPsec, SSH, TLS, HTTPS].**

Application Note 28

This requirement ensures all communications between components of a distributed TOE is protected through the use of an encrypted communications channel. The data passed in this trusted communication channel are encrypted as defined the protocol chosen in the first selection. The ST author selects the mechanism or mechanisms supported by the TOE, and *then ensures that the detailed protocol requirements in Appendix B corresponding to their selection are included in the ST.*

5.7.2 Protection of TSF Data (Extended – FPT_SKP_EXT)

5.7.2.1 FPT_SKP_EXT.1 Protection of TSF Data (for reading of all symmetric keys)

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

FPT_SKP_EXT.1.1 The TSF shall prevent reading of all pre-shared keys, symmetric keys, and private keys.

Application Note 29

The intent of the requirement is that an administrator is unable to read or view the identified keys (stored or ephemeral) through "normal" interfaces. While it is understood that the administrator could directly read memory to view these keys, do so is not a trivial task and may require substantial work on the part of an administrator. Since the administrator is considered a trusted agent, it is assumed they would not endeavour in such an activity.

5.7.3 Protection of Administrator Passwords (Extended – FPT_APW_EXT)

5.7.3.1 FPT_APW_EXT.1 Protection of Administrator Passwords

FPT_APW_EXT.1	Protection of Administrator Passwords

FPT_APW_EXT.1.1 The TSF shall store passwords in non-plaintext form.

FPT_APW_EXT.1.2 The TSF shall prevent the reading of plaintext passwords.

Application Note 30

The intent of the requirement is that raw password authentication data are not stored in the clear, and that no user or administrator is able to read the plaintext password through "normal" interfaces. An all-powerful administrator of course could directly read memory to capture a password but is trusted not to do so.

5.7.4 TSF testing (Extended – FPT_TST_EXT)

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

(For this component, selection-based requirements exist in Appendix B)

5.7.4.1 FPT_TST_EXT.1 TSF Testing (Extended)

FPT_TST_EXT.1	**TSF testing**

FPT_TST_EXT.1.1 The TSF shall run a suite of the following self-tests [selection: *during initial start-up (on power on), periodically during normal operation, at the request of the authorised user, at the conditions [assignment: conditions under which self-tests should occur]*] to demonstrate the correct operation of the TSF: [assignment: *list of self-tests run by the TSF during initial start-up*].

Application Note 31

It is expected that self-tests are carried out during initial start-up (on power on). Other options shall only be used if the developer can justify why they are not carried out during initial start-up. It is expected that at least self-tests for verification of the integrity of the firmware and software as well as for the correct operation of cryptographic functions necessary to fulfil the SFRs will be performed. If not all self-test are performed during start-up multiple iterations of this SFR shall be used with the appropriate options selected. In future versions of this cPP the suite of self-tests will be required to contain at least mechanisms for measured boot including self-tests of the components which perform the measurement.

Application Note 32

If certificates are used by the self-test mechanism (e.g. for verification of signatures for integrity verification), certificates are validated in accordance with FIA_X509_EXT.1 and should be selected in FIA_X509_EXT.2.1. Additionally, FPT_TST_EXT.2.1 must be included in the ST.

5.7.5 Trusted Update (FPT_TUD_EXT)

Failure by the Security Administrator to verify that updates to the system can be trusted may lead to compromise of the entire system. To establish trust in the source of the updates, the system can provide cryptographic mechanisms and procedures to procure the update, check the update cryptographically through the TOE-provided digital signature mechanism, and install the update on the system. While there is no requirement that this process be completely automated, administrative guidance documentation will detail any procedures that must be performed manually, as well as the manner in which the administrator ensures that the signature on the update is valid.

(For this family, selection-based requirements exist in Appendix B)

5.7.5.1 FPT_TUD_EXT.1 Trusted Update

FPT_TUD_EXT.1	**Trusted update**

FPT_TUD_EXT.1.1 The TSF shall provide <u>*Security Administrators*</u> the ability to query the currently executed version of the TOE firmware/software as well as the most recently installed version of the TOE firmware/software.

Application Note 33

The version currently running (being executed) may not be the version most recently installed. For instance, maybe the update was installed but the system requires a reboot before this update will run. Therefore, it needs to be clear that the query should indicate both the most recently executed version as well as the most recently installed update.

FPT_TUD_EXT.1.2 The TSF shall provide Security Administrators the ability to manually initiate updates to TOE firmware/software and [selection: *support automatic updates, no other update mechanism*].

FPT_TUD_EXT.1.3 The TSF shall provide means to authenticate firmware/software updates to the TOE using a digital signature mechanism prior to installing those updates.

Application Note 34

The digital signature mechanism referenced in FPT_TUD_EXT.1.3 is one of the algorithms specified in FCS_COP.1(2).

Application Note 35

If certificates are used by the update verification mechanism, certificates are validated in accordance with FIA_X509_EXT.1 and should be selected in FIA_X509_EXT.2.1. Additionally, FPT_TUD_EXT.2.1 must be included in the ST.

Application Note 36

"Update" in the context of this SFR refers to the process of replacing a non-volatile, system resident software component with another. The former is referred to as the NV image, and the latter is the update image. While the update image is typically newer than the NV image, this is not a requirement. There are legitimate cases where the system owner may want to rollback a component to an older version (e.g. when the component manufacturer releases a faulty update, or when the system relies on an undocumented feature no longer present in the update). Likewise, the owner may want to update with the same version as the NV image to recover from faulty storage.

All discrete software components (e.g. applications, drivers, kernel, firmware) of the TSF, should be digitally signed by the corresponding manufacturer and subsequently verified by the mechanism performing the update. Since it is recognized that components may be signed by different manufacturers, it is essential that the update process verify that both the update and NV images were produced by the same manufacturer (e.g. by comparing public keys) or signed by legitimate signing keys (e.g. successful verification of certificates when using X.509 certificates).

5.7.6 Time stamps (FPT_STM)

5.7.6.1 FPT_STM.1 Reliable Time Stamps

FPT_STM.1	Reliable Time Stamps

FPT_STM.1.1 The TSF shall be able to provide reliable time stamps.

Application Note 37

The TSF does not provide reliable information about the current time at the TOE's location by itself, but depends on external time and date information, either provided manually by the administrator or through the use of an NTP server. The term 'reliable time stamps' refers to the strict use of the time and date information, that is provided externally, and the logging of all changes to the time settings including information about the old and new time. With this information the real time for all audit data can be calculated.

5.8 TOE Access (FTA)

5.8.1 TSF-initiated Session Locking (Extended – FTA_SSL_EXT)

5.8.1.1 FTA_SSL_EXT.1 TSF-initiated Session Locking

FTA_SSL_EXT.1	**TSF-initiated Session Locking**

FTA_SSL_EXT.1.1 The TSF shall, for local interactive sessions, [selection:

- *lock the session - disable any activity of the user's data access/display devices other than unlocking the session, and requiring that the administrator re-authenticate to the TSF prior to unlocking the session;*
- *terminate the session*]

after a Security Administrator-specified time period of inactivity.

5.8.2 Session locking and termination (FTA_SSL)

5.8.2.1 FTA_SSL.3 TSF-initiated Termination

FTA_SSL.3	**TSF-initiated Termination**

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

5.8.2.2 FTA_SSL.4 User-initiated Termination

FTA_SSL.4	**User-initiated Termination**

FTA_SSL.4.1 Refinement: The TSF shall allow **Administrator**-initiated termination of the **Administrator**'s own interactive session.

5.8.3 TOE access banners (FTA_TAB)

5.8.3.1 FTA_TAB.1 Default TOE Access Banners

FTA_TAB.1	**Default TOE Access Banners**

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

Application Note 38

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

5.9 Trusted path/channels (FTP)

5.9.1 Trusted Channel (FTP_ITC)

5.9.1.1 FTP_ITC.1 Inter-TSF trusted channel (Refined)

FTP_ITC.1	**Inter-TSF trusted channel**

FTP_ITC.1.1 The TSF shall be **capable of using [selection: *IPsec, SSH, TLS, HTTPS*] to** provide a trusted communication channel between itself and **authorized IT entities supporting the following capabilities: audit server, [selection: *authentication server, assignment: [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 **<u>the TSF, or the authorized IT entities</u>** to initiate communication via the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for [assignment: *list of **services for which the TSF is able to initiate communications***].

Application Note 39

The intent of the above requirement is to provide a means by which a cryptographic protocol may be used to protect external communications with authorized IT entities that the TOE interacts with to perform its functions. The TOE shall be capable of providing protection (by one of the listed protocols) at least for communications with the server that collects the audit information. If it communicates with an authentication server (e.g., RADIUS), then the ST author chooses "authentication server" in FTP_ITC.1.1 and this connection must be capable of being protected by one of the listed protocols. If other authorized IT entities (e.g., NTP server) are protected, the ST author makes the appropriate assignments (for those entities) and selections (for the protocols that are used to protect those connections). The ST author selects the mechanism or mechanisms supported by the TOE, and then ensures that the *detailed protocol requirements in Appendix B corresponding to their selection are included in the ST.*

While there are no requirements on the party initiating the communication, the ST author lists in the assignment for FTP_ITC.1.3 the services for which the TOE can initiate the communication with the authorized IT entity.

The requirement implies that not only are communications protected when they are initially established, but also on resumption after an outage. It may be the case that some part of the TOE setup involves manually setting up tunnels to protect other communication, and if after an outage the TOE attempts to re-establish the communication automatically with (the necessary) manual intervention, there may be a window created where an attacker might be able to gain critical information or compromise a connection.

5.9.2 Trusted Path (FTP_TRP)

5.9.2.1 FTP_TRP.1 Trusted Path (Refinement)

FTP_TRP.1	**Trusted Path**

FTP_TRP.1.1 The TSF shall **be capable of using [selection: *IPsec, SSH, TLS, HTTPS*] to** provide a trusted communication channel between itself and **authorized IT entities supporting the following capabilities: audit server, [selection: *authentication server, assignment: [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_TRP.1.2 The TSF shall permit **remote administrators** to initiate communication via the trusted path.

FTP_TRP.1.3 The TSF shall require the use of the trusted path for **initial administrator authentication and all remote administration actions**.

Application Note 40

This requirement ensures that authorized remote administrators initiate all communication with the TOE via a trusted path, and that all communication with the TOE by remote administrators is performed over this path. The data passed in this trusted communication channel are encrypted as defined by the protocol chosen in the first selection. The ST author selects the mechanism or mechanisms supported by the TOE, and then ensures that the detailed protocol requirements in Appendix B corresponding to their selection are included in the ST.

5.10 Firewall (FFW)

5.10.1 Stateful Traffic Filter Firewall (FFW_RUL_EXT)

5.10.1.1 FFW_RULEXT.1 Stateful Traffic Filtering

FFW_RULEXT.1	**Stateful Traffic Filtering**

FFW_RUL_EXT.1.1 The TSF shall perform Stateful Traffic Filtering on network packets processed by the TOE.

Application Note 41

This element identifies the policy (Stateful Traffic Filtering) that is applied to the network packets that are processed at the TOE's interfaces. Every packet that is received at a TOE's interface either has the ruleset that expresses this policy applied, or it is determined that the packet belongs to an established connection. The remaining elements in this component provide the details of the policy.

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

It is important to note that the TOE, which also includes the underlying platform, cannot permit network packets to flow unless the ruleset contains a rule that permits the flow, or the packet is deemed to belong to an established connection that has been permitted to flow. This principle must hold true during TOE startup, and upon failures the TOE may encounter.

FFW_RUL_EXT.1.2 The TSF shall allow the definition of Stateful Traffic Filtering rules using the following network protocol fields:

- ICMPv4
  - Type
  - Code
- ICMPv6
  - Type
  - Code
- IPv4
  - Source address
  - Destination Address
  - Transport Layer Protocol
- IPv6
  - Source address
  - Destination Address
  - Transport Layer Protocol
  - [selection: *IPv6 Extension header type [assignment: list of fields in IPv6 extension header], no other field*]
- TCP
  - Source Port
  - Destination Port
- UDP
  - Source Port
  - Destination Port

- and distinct interface.

Application Note 42

This element identifies the various attributes that are applicable when constructing rules to be enforced by this requirement – the applicable interface is a property of the TOE and the rest of the identified attributes are defined in the associated RFCs. Note that the 'Transport Layer Protocol' is the IPv4/IPv6 field that identifies the applicable protocol, such as TCP, UDP, ICMP, or GRE. IPv6 extension headers are defined in RFC 2460 and the ST author may specify which fields within each supported extension header, if any may be used as attributes in the construction of an inspection rule. Also, 'Interface' identified above is the external port where the applicable network traffic was received or will be sent.

FFW_RUL_EXT.1.3 The TSF shall allow the following operations to be associated with Stateful Traffic Filtering rules: permit or drop with the capability to log the operation.

Application Note 43

This element defines the operations that can be associated with rules used to match network traffic. Note that the data to be logged is identified in the Security Audit requirements in Table 1.

FFW_RUL_EXT.1.4 The TSF shall allow the Stateful Traffic Filtering rules to be assigned to each distinct network interface.

Application Note 44

This element identifies where rules can be assigned. Specifically, a conforming TOE must be able to assign filtering rules to each of its available and distinct network interfaces that handle layer 3 and 4 network traffic. A distinct network interface can be physical or logical but it does not necessarily required to be visible from the network perspective (e.g. it does not need to have an IP address assigned to it).

Note that there could be a separate ruleset for each interface or alternately a shared ruleset that somehow associates rules with specific interfaces.

FFW_RUL_EXT.1.5 The TSF shall:

a) accept a network packet without further processing of Stateful Traffic Filtering rules if it matches an allowed established session for the following protocols: TCP, UDP, [selection: *ICMP, no other protocols*] based on the following network packet attributes:
    1. TCP: source and destination addresses, source and destination ports, sequence number, Flags;
    2. UDP: source and destination addresses, source and destination ports;
    3. [selection: *'ICMP: source and destination addresses, type, [selection: code, [assignment: list of matching attributes]]', no other protocols*].

b) Remove existing traffic flows from the set of established traffic flows based on the following: [selection: *session inactivity timeout, completion of the expected information flow*].

Application Note 45

This element requires that the protocols be identified for which the TOE can determine and manage the state such that sessions can be established and are used to make traffic flow decisions as opposed to fully processing the configured rules. This element also requires that applicable attributes used to determine whether a network packet matches and established session are identified.

If ICMP is selected as a protocol the source and destination addresses are required to be considered when determining if a packet belongs to an established "connection". The type and code attributes may be used to provide a more robust capability in determining whether an ICMP packet is what is expected in an established connection flow. For example, one would not expect echo replies to be part of a flow if an echo request had not been received. The open assignment in the selection for ICMP attributes is left for implementations that may use IPv6 attributes.

Item b) in this element requires specification of how the firewall can determine that established information flows should be removed from the set of established information flows by observing events such as the termination of a TCP session initiated by either endpoint with FIN flags in the TCP packet. If protocols are handled differently, it is expected that the ST would identify those differences.

[OPTIONAL] FFW_RUL_EXT.1.6 The TSF shall dynamically define rules or establish sessions allowing network traffic to flow for the following network protocols [selection: *FTP, SIP, H.323: [assignment: other supported protocols], no other protocols*].

Application Note 46

This element requires the specification of more complex protocols that require the firewall to allow network traffic flow even though an existing rule does not explicitly allow the flow. For example, the FTP protocol requires both a control connection and a data connection if a user is to transfer files. While there are well-known ports involved, port 21 (control port on FTP server) and port 20 (data port on server in active mode), there are random ports > 1023 used on the client side. In passive mode, the FTP server may use a random port >1023 instead of port 20. The data connection is initiated by the client in passive mode, and imitated by the FTP server in active mode.

For these types of protocols, the establishment of a "new" connection is allowed, even though the ruleset may appear to deny it (e.g., since a rule cannot predict which random port will be used by the client or potentially the server, the default rule to deny may appear to apply). The TSF could create a dynamic rule that governs the traffic flow, or the TSF could implicitly allow the new connection to be established based on expectations of the protocol implementation as specified in the RFC or equivalent standard.

It is important to note that there is no expectation that any network packets be inspected beyond layer 4 (TCP/UDP). This requirement simply requires that the ST author specify the conditions under which a a rule is dynamically inserted into the firewall to allow expected connections with unpredictable UDP/TCP ports to correctly be established.

If the ST Author includes additional protocols they must identify the RFC or equivalent standard that specifies the behavior of the protocol, as is done for FTP above.

FFW_RUL_EXT.1.7 The TSF shall enforce the following default Stateful Traffic Filtering rules on all network traffic:

1. The TSF shall drop and be capable of [selection: *counting, logging*] packets which are invalid fragments;
2. The TSF shall drop and be capable of [selection: *counting, logging*] fragmented packets which cannot be re-assembled completely;
3. The TSF shall drop and be capable of logging packets where the source address of the network packet is defined as being on a broadcast network;
4. The TSF shall drop and be capable of logging packets where the source address of the network packet is defined as being on a multicast network;
5. The TSF shall drop and be capable of logging network packets where the source address of the network packet does not belong to the networks associated with the network interface where the network packet was received;
6. The TSF shall drop and be capable of logging network packets where the source address of the network packet is defined as being a loopback address;
7. The TSF shall drop and be capable of logging network packets where the source or destination address of the network packet is defined as being unspecified (i.e. 0.0.0.0) or an address “reserved for future use” (i.e. 240.0.0.0/4) as specified in RFC 5735 for IPv4;
8. The TSF shall drop and be capable of logging network packets where the source or destination address of the network packet is defined as an “unspecified address” or an address “reserved for future definition and use” (i.e. unicast addresses not in this address range: 2000::/3) as specified in RFC 3513 for IPv6;
9. The TSF shall drop and be capable of logging network packets with the IP options: Loose Source Routing, Strict Source Routing, or Record Route specified; and
10. [selection: *[assignment: other default rules enforced by the TOE], no other rules*].

Application Note 47

Note that these rules must apply by default, without requiring configuration.

FFW_RUL_EXT.1.8 The TSF shall be capable of dropping and logging according to the following rules:

1. The TSF shall reject and be capable of logging network packets where the source address of the network packet is equal to the address of the network interface where the network packet was received;
2. The TSF shall reject and be capable of logging network packets where the source or destination address of the network packet is a link-local address;

Application Note 48

Note that these rules may be configured (as opposed to applying by default).

FFW_RUL_EXT.1.9 , The TSF shall process the applicable Stateful Traffic Filtering rules in an administratively defined order.

Application Note 49

This element requires that an administrator is able to define the order in which configured filtering rules are processed for matches. The filtering rules are only applicable when an allowed session has not been established or a dynamic rule has been created.

FFW_RUL_EXT.1.10 The TSF shall deny packet flow if a matching rule is not identified.

Application Note 50

This element requires that, except when a packet is part of an established session, the behavior is always to deny network traffic when no rules apply and no other operations are required, though they are not necessarily prohibited.

FFW_RUL_EXT.1.11 The TSF shall be capable of limiting an administratively configured number of *half-open TCP connections: [selection: with a common destination IP address, with a common destination IP address and TCP port tuple]*.

Application Note 51

A half-open TCP connection is one that has not completed the full three-way handshake as defined in RFC 793. Incomplete TCP connections i.e. those that have completed the SYN and SYN-ACK portions of the three-way handshake consume valuable resources in end hosts and stateful traffic filtering devices in the traffic path and, in sufficient volume, can lead to a denial of service condition. To protect itself, and any targeted protected services, compliant TOEs shall be capable of limiting the number of half-open TCP connections targeted at a specific destination IP address and port number. Optionally, the ST author may also define additional methods of policing i.e. a maximum number of half-open connections for a specific client (i.e. common source IP address).

6. Security Assurance Requirements

This cPP identifies the Security Assurance Requirements (SARs) to frame the extent to which the evaluator assesses the documentation applicable for the evaluation and performs independent testing.

This section lists the set of SARs from CC part 3 that are required in evaluations against this cPP. Individual Evaluation Activities to be performed are specified in [SD].

The general model for evaluation of TOEs against STs written to conform to this cPP is as follows: after the ST has been approved for evaluation, the ITSEF will obtain the TOE, supporting environmental IT (if required), and the administrative/user guides for the TOE. The ITSEF is expected to perform actions mandated by the Common Evaluation Methodology (CEM) for the ASE and ALC SARs. The ITSEF also performs the Evaluation Activities contained within the SD, which are intended to be an interpretation of the other CEM assurance requirements as they apply to the specific technology instantiated in the TOE. The Evaluation Activities that are captured in the SD also provide clarification as to what the developer needs to provide to demonstrate the TOE is compliant with the cPP.

The TOE security assurance requirements are identified in Table 2.

Assurance Class	**Assurance Components**
Security Target (ASE)	Conformance claims (ASE_CCL.1)
	Extended components definition (ASE_ECD.1)
	ST introduction (ASE_INT.1)
	Security objectives for the operational environment (ASE_OBJ.1)
	Stated security requirements (ASE_REQ.1)
	Security Problem Definition (ASE_SPD.1)
	TOE summary specification (ASE_TSS.1)
Development (ADV)	Basic functional specification (ADV_FSP.1)
Guidance documents (AGD)	Operational user guidance (AGD_OPE.1)
	Preparative procedures (AGD_PRE.1)
Life cycle support (ALC)	Labeling of the TOE (ALC_CMC.1)
	TOE CM coverage (ALC_CMS.1)
Tests (ATE)	Independent testing – sample (ATE_IND.1)
Vulnerability assessment (AVA)	Vulnerability survey (AVA_VAN.1)

Table 2: Security Assurance Requirements

6.1 ASE: Security Target

The ST is evaluated as per ASE activities defined in the CEM. In addition, there may be Evaluation Activities specified within the SD that call for necessary descriptions to be included in the TSS that are specific to the TOE technology type.

The SFRs in this cPP allow for conformant implementations to incorporate a wide range 0f acceptable key management approaches as long as basic principles are satisfied. Given the criticality of the key management scheme, this cPP requires the developer to provide a detailed description of their key management implementation. This information can be submitted as an appendix to the ST and marked proprietary, as this level of detailed information is not expected to be made publicly available. See Appendix E for details on the expectation of the developer's Key Management Description.

In addition, if the TOE includes a random bit generator Appendix D provides a description of the information expected to be provided regarding the quality of the entropy.

ASE_TSS.1.1C Refinement: The TOE summary specification shall describe how the TOE meets each SFR**, including required supplementary information on Entropy.**

6.2 ADV: Development

The design information about the TOE is contained in the guidance documentation available to the end user as well as the TSS portion of the ST, and any required supplementary information required by this cPP that is not to be made public.

6.2.1 Basic Functional Specification (ADV_FSP.1)

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

The Evaluation Activities in the SD are associated with the applicable SFRs; since these are directly associated with the SFRs, the tracing in element ADV_FSP.1.2D is implicitly already done and no additional documentation is necessary.

6.3 AGD: Guidance Documentation

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

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

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

- instructions to provide a protected administrative capability.

Guidance pertaining to particular security functionality must also be provided; requirements on such guidance are contained in the Evaluation Activities specified in the SD.

6.3.1 Operational User Guidance (AGD_OPE.1)

The operational user guidance does not have to be contained in a single document. Guidance to users, administrators and application developers can be spread among documents or web pages.

The developer should review the Evaluation Activities contained in the SD to ascertain the specifics of the guidance that the evaluator will be checking for. This will provide the necessary information for the preparation of acceptable guidance.

6.3.2 Preparative Procedures (AGD_PRE.1)

As with the operational guidance, the developer should look to the Evaluation Activities to determine the required content with respect to preparative procedures.

6.4 Class ALC: Life-cycle Support

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

6.4.1 Labelling of the TOE (ALC_CMC.1)

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

6.4.2 TOE CM Coverage (ALC_CMS.1)

Given the scope of the TOE and its associated evaluation evidence requirements, the evaluator performs the CEM work units associated with ALC_CMC.1.

6.5 Class ATE: Tests

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

6.5.1 Independent Testing – Conformance (ATE_IND.1)

Testing is performed to confirm the functionality described in the TSS as well as the operational guidance (includes “evaluated configuration” instructions). The focus of the testing is to confirm that the requirements specified in Section 5 are being met. The Evaluation Activities in the SD identify the specific testing activities necessary to verify compliance with the SFRs. The evaluator produces a test report documenting the plan for and results of testing, as well as coverage arguments focused on the platform/TOE combinations that are claiming conformance to this cPP.

6.6 Class AVA: Vulnerability Assessment

For the first generation of this cPP, the iTC is expected to survey open sources to discover what vulnerabilities have been discovered in these types of products and provide that content into the AVA_VAN discussion. In most cases, these vulnerabilities will require sophistication beyond that of a basic attacker. This information will be used in the development of future protection profiles.

6.6.1 Vulnerability Survey (AVA_VAN.1)

Appendix A in [SD] provides a guide to the evaluator in performing a vulnerability analysis.

A. Optional Requirements

As indicated in the introduction to this cPP, the baseline requirements (those that must be performed by the TOE) are contained in the body of this cPP. Additionally, there are two other types of requirements specified in Appendices A and B.

The first type (in this Appendix) is requirements that can be included in the ST, but do not have to be in order for a TOE to claim conformance to this cPP. The second type (in Appendix B) is requirements based on selections in the body of the cPP: if certain selections are made, then additional requirements in that appendix will need to be included in the body of the ST (e.g., cryptographic protocols selected in a trusted channel requirement).

A.1 Audit Events for Optional SFRs

Requirement	Auditable Events	Additional Audit Record Contents
FAU_STG_EXT.2	None.	None.
FMT_MOF.1(1)/Audit	None.	None.
FMT_MOF.1(2)/Audit	None.	None.
FMT_MOF.1(1)/AdminAct	None.	None.
FMT_MOF.1(2)/AdminAct	None.	None.
FMT_MOF.1(1)/LocSpace	None.	None.
FMT_MTD.1/AdminAct	None.	None.
FPT_FLS.1/Local Audit Storage Space Full	None.	None.

Table 3: TOE Optional SFRs and Auditable Events

A.2 Security Audit (FAU)

A.2.1 Security audit event storage (Extended – FAU_STG_EXT)

A.2.1.1 FAU_ STG_EXT.2 Counting lost audit data

FAU_STG_EXT.2	Counting lost audit data

FAU_STG_EXT.2.1 The TSF shall provide information about the number of [selection: *dropped, overwritten, assignment: other information*] audit records in the case where the local storage has been filled and the TSF takes one of the actions defined in FAU_STG_EXT.1.3.

Application Note 52

This option should be chosen if the TOE supports this functionality.

In case the local storage for audit records is cleared by the administrator, the counters associated with the selection in the SFR should be reset to their initial value (most likely to 0). The guidance documentation shall contain a warning for the administrator about the loss of audit data when he clears the local storage for audit records.

A.3 Security Management (FMT)

A.3.1 Management of functions in TSF (FMT_MOF)

A.3.1.1 FMT_MOF.1 Management of security functions behaviour

FMT_MOF.1(1)/Audit	**Management of security functions behaviour**

FMT_MOF.1.1(1)/Audit The TSF shall restrict the ability to *determine the behaviour of, modify the behaviour* of the functions *transmission of audit data to an external IT entity* to *Security Administrators*.

Application Note 53

FMT_MOF.1(1)/Audit should only be chosen if the transmission protocol for transmission of audit data to an external IT entity as defined in FAU_STG_EXT.1.1 is configurable.

FMT_MOF.1(2)/Audit	**Management of security functions behaviour**

FMT_MOF.1.1(2)/Audit The TSF shall restrict the ability to determine the *behaviour of, modify the behaviour* of the functions *handling of audit data* to *Security Administrators*.

Application Note 54

FMT_MOF.1(2)/Audit should only be chosen if the handling of audit data is configurable. The term ‘handling of audit data’ refers to the different options for selection and assignments in SFRs FAU_STG_EXT.1.2, FAU_STG_EXT.1.3 and FAU_STG_EXT.2.

FMT_MOF.1(1)/AdminAct	**Management of security functions behaviour**

FMT_MOF.1.1(1)/AdminAct The TSF shall restrict the ability to *modify the behaviour* of the functions *TOE Security Functions* to *Security Administrators*.

FMT_MOF.1(2)/AdminAct	**Management of security functions behaviour**

FMT_MOF.1.1(2)/AdminAct The TSF shall restrict the ability to enable, disable of the functions *initiate starting and stopping services* to *Security Administrators*.

FMT_MOF.1/LocSpace	**Management of security functions behaviour**

FMT_MOF.1.1/LocSpace The TSF shall restrict the ability to *determine the behaviour of, modify the behaviour* of the functions *audit functionality when Local Audit Storage Space is full* to *Security Administrators*.

A.3.2 Management of TSF data (FMT_MTD)

A.3.2.1 FMT_MTD.1/AdminAct Management of TSF data

FMT_MTD.1/AdminAct	**Management of TSF data**

FMT_MTD.1.1/AdminAct The TSF shall restrict the ability to *modify, delete, generate/import* the *cryptographic keys* to *Security Administrators*.

A.4 Protection of the TSF (FPT)

A.4.1 Fail Secure (FPT_FLS)

A.4.1.1 FPT_FLS.1/LocSpace Failure with preservation of secure state

FPT_FLS.1/LocSpace	**Failure with preservation of secure state**

FPT_FLS.1.1/LocSpace The TSF shall preserve a secure state when the following types of failures occur: *Local Storage Space for audit data is full.*

Application Note 55

Preserving a secure state in the sense of this SFR means to stop all security functions as long as there is no more local storage space available.

B. Selection-Based Requirements

As indicated in the introduction to this PP, the baseline requirements (those that must be performed by the TOE or its underlying platform) are contained in the body of this PP. There are additional requirements based on selections in the body of the PP: if certain selections are made, then additional requirements below will need to be included.

B.1 Audit Events for Selection-Based SFRs

Requirement	Auditable Events	Additional Audit Record Contents
FCS_HTTPS_EXT.1	Failure to establish a HTTPS Session.	Reason for failure
FCS_IPSEC_EXT.1	Failure to establish an IPsec SA.	Reason for failure
FCS_SSHC_EXT.1	Failure to establish an SSH session	Reason for failure
	Successful SSH rekey	Non-TOE endpoint of connection (IP Address)
FCS_SSHS_EXT.1	Failure to establish an SSH session	Reason for failure
	Successful SSH rekey	Non-TOE endpoint of connection (IP Address)
FCS_TLSC_EXT.1	Failure to establish a TLS Session	Reason for failure
FCS_TLSS_EXT.1	Failure to establish a TLS Session	Reason for failure
FMT_MOF.1(1)/TrustedUpdate	None.	None.
FPT_TST_EXT.2	Failure of self-test	Reason for failure (including identifier of invalid certificate)
FPT_TUD_EXT.2	Failure of update	Reason for failure (including identifier of invalid certificate)

Table 4: Selection-Dependent SFRs and Auditable Events

B.2 Cryptographic Support (FCS)

B.2.1 Cryptographic Protocols (Extended – FCS_HTTPS_EXT, FCS_ IPSEC_EXT, FCS_SSHC_EXT, FCS_SSHS_EXT, FCS_TLSC_EXT, FCS_TLSS_EXT)

B.2.1.1 FCS_HTTPS_EXT.1 HTTPS Protocol

FCS_HTTPS_EXT.1	HTTPS Protocol

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

Application Note 56

The ST author must provide enough detail to determine how the implementation is complying with the standard(s) identified; this can be done either by adding elements to this component, or by additional detail in the TSS.

FCS_HTTPS_EXT.1.2 The TSF shall implement HTTPS using TLS.

FCS_HTTPS_EXT.1.3 The TSF shall [selection: *not establish the connection, request authorization to establish the connection, no other action*] if the peer certificate is deemed invalid.

Application Note 57

Validity is determined by the certificate path, the expiration date, and the revocation status in accordance with RFC 5280.

B.2.1.2 FCS_IPSEC_EXT.1 IPsec Protocol

FCS_IPSEC_EXT.1	IPsec Protocol

FCS_IPSEC_EXT.1.1 The TSF shall implement the IPsec architecture as specified in RFC 4301.

Application Note 58

RFC 4301 calls for an IPsec implementation to protect IP traffic through the use of a Security Policy Database (SPD). The SPD is used to define how IP packets are to be handled: PROTECT the packet (e.g., encrypt the packet), BYPASS the IPsec services (e.g., no encryption), or DISCARD the packet (e.g., drop the packet). The SPD can be implemented in various ways, including router access control lists, firewall rulesets, a “traditional” SPD, etc. Regardless of the implementation details, there is a notion of a “rule” that a packet is “matched” against and a resulting action that takes place.

While there must be a means to order the rules, a general approach to ordering is not mandated, as long as the SPD can distinguish the IP packets and apply the rules accordingly. There may be multiple SPDs (one for each network interface), but this is not required.

FCS_IPSEC_EXT.1.2 The TSF shall have a nominal, final entry in the SPD that matches anything that is otherwise unmatched, and discards it.

FCS_IPSEC_EXT.1.3 The TSF shall implement *transport mode and [selection: tunnel mode, no other mode]*.

FCS_IPSEC_EXT.1.4 The TSF shall implement the IPsec protocol ESP as defined by RFC 4303 using the cryptographic algorithms AES-CBC-128, AES-CBC-256 (both specified by RFC 3602) and [selection: *AES-GCM-128 (specified in RFC 4106), AES-GCM-256 (specified in RFC 4106), no other algorithms*] together with a Secure Hash Algorithm (SHA)-based HMAC.

FCS_IPSEC_EXT.1.5 The TSF shall implement the protocol: [selection:

- *IKEv1 as defined in RFCs 2407, 2408, 2409, RFC 4109, [selection: no other RFCs for extended sequence numbers, RFC 4304 for extended sequence numbers], and [selection: no other RFCs for hash functions, RFC 4868 for hash functions];*
- *IKEv2 as defined in RFC 5996 and [selection: with no support for NAT traversal, with mandatory support for NAT traversal as specified in RFC 5996, section 2.23)], and [selection: no other RFCs for hash functions, RFC 4868 for hash functions]*

].

Application Note 59

If the TOE implements SHA-2 hash algorithms for IKEv1 or IKEv2, the ST author shall select RFC 4868. If the ST author selects IKEv1, FCS_IPSEC_EXT.1.15 must also be included in the ST. IKEv2 will be required for those TOEs entering evaluation after Quarter 3, 2016.

FCS_IPSEC_EXT.1.6 The TSF shall ensure the encrypted payload in the [selection: *IKEv1, IKEv2*] protocol uses the cryptographic algorithms AES-CBC-128, AES-CBC-256 as specified in RFC 3602 and [selection: *AES-GCM-128, AES-GCM-256 as specified in RFC 5282, no other algorithm*].

Application Note 60

AES-GCM-128 and AES-GCM-256 may only be selected if IKEv2 is also selected, as there is no RFC defining AES-GCM for IKEv1.

FCS_IPSEC_EXT.1.7 The TSF shall ensure that [selection:

- *IKEv1 Phase 1 SA lifetimes can be configured by an Security Administrator based on [selection:*
    - *number of packets/number of bytes;*
    - *length of time, where the time values can configured within [assignment: integer range including 24] hours;*

  ];
- *IKEv2 SA lifetimes can be configured by an Security Administrator based on [selection:*
    - *number of packets/number of bytes;*
    - *length of time, where the time values can configured within [assignment: integer range including 24] hours*

  ]

].

Application Note 61

The ST author chooses either the IKEv1 requirements or IKEv2 requirements (or both, depending on the selection in FCS_IPSEC_EXT.1.5). The ST author chooses either packet/volume-based lifetimes or time-based lifetimes. This requirement must be accomplished by providing Security Administrator-configurable lifetimes (with appropriate instructions in documents mandated by AGD_OPE). Hardcoded limits are not acceptable. In general, instructions for setting the parameters of the implementation, including lifetime of the SAs, should be included in the operational guidance generated for AGD_OPE.

FCS_IPSEC_EXT.1.8 The TSF shall ensure that [selection:

- *IKEv1 Phase 2 SA lifetimes can be configured by a Security Administrator based on [selection:*
  - *number of packets/number of bytes;*
  - *length of time, where the time values can be configured within [assignment: integer range including 8] hours;*

  ];

- *IKEv2 Child SA lifetimes can be configured by a Security Administrator based on [selection:*
  - *number of packets/number of bytes;*
  - *length of time, where the time values can be configured within [assignment: integer range including 8] hours;*

  ]

].

Application Note 62

The ST author chooses either the IKEv1 requirements or IKEv2 requirements (or both, depending on the selection in FCS_IPSEC_EXT.1.5). The ST author chooses either packet/volume-based lifetimes or time-based lifetimes. This requirement must be accomplished by providing Security Administrator-configurable lifetimes (with appropriate instructions in documents mandated by AGD_OPE). Hardcoded limits are not acceptable. In general, instructions for setting the parameters of the implementation, including lifetime of the SAs, should be included in the operational guidance generated for AGD_OPE.

FCS_IPSEC_EXT.1.9 The TSF shall generate the secret value x used in the IKE Diffie-Hellman key exchange (“x” in g^x mod p) using the random bit generator specified in FCS_RBG_EXT.1, and having a length of at least [assignment: *(one or more) number(s) of bits that is at least twice the security strength of the negotiated Diffie-Hellman group*] bits.

Application Note 63

For DH groups 19 and 20, the "x" value is the point multiplier for the generator point G.

Since the implementation may allow different Diffie-Hellman groups to be negotiated for use in forming the SAs, the assignment in FCS_IPSEC_EXT.1. may contain multiple values. For each DH group supported, the ST author consults Table 2 in NIST SP 800-57

"Recommendation for Key Management –Part 1: General" to determine the security strength ("bits of security") associated with the DH group. Each unique value is then used to fill in the assignment. For example, suppose the implementation supports DH group 14 (2048-bit MODP) and group 20 (ECDH using NIST curve P-384). From Table 2, the bits of security value for group 14 is 112, and for group 20 it is 192.

FCS_IPSEC_EXT.1.10 The TSF shall generate nonces used in [selection: *IKEv1, IKEv2*] exchanges of length [selection:

- *[assignment: security strength associated with the negotiated Diffie-Hellman group];*
- *at least 128 bits in size and at least half the output size of the negotiated pseudorandom function (PRF) hash*

  ] .

Application Note 64

The ST author must select the second option for nonce lengths if IKEv2 is also selected (as this is mandated in RFC 5996). The ST author may select either option for IKEv1.

For the first option for nonce lengths, since the implementation may allow different Diffie-Hellman groups to be negotiated for use in forming the SAs, the assignment in FCS_IPSEC_EXT.1. may contain multiple values. For each DH group supported, the ST author consults Table 2 in NIST SP 800-57 "Recommendation for Key Management –Part 1: General" to determine the security strength ("bits of security") associated with the DH group. Each unique value is then used to fill in the assignment. For example, suppose the implementation supports DH group 14 (2048-bit MODP) and group 20 (ECDH using NIST curve P-384). From Table 2, the bits of security value for group 14 is 112, and for group 20 it is 192.

Because nonces may be exchanged before the DH group is negotiated, the nonce used should be large enough to support all TOE-chosen proposals in the exchange.

FCS_IPSEC_EXT.1.11 The TSF shall ensure that all IKE protocols implement DH Groups 14 (2048-bit MODP), and [selection: *19 (256-bit Random ECP), 5 (1536-bit MODP), 24 (2048-bit MODP with 256-bit POS), 20 (384-bit Random ECP), no other DH groups*].

Application Note 65

The selection is used to specify additional DH groups supported. This applies to IKEv1 and IKEv2 exchanges. For products entering into evaluation after Quarter 3, 2015, DH Group 19 (256-bit Random ECP) and DH Group 20 (384-bit Random ECP) will be required. It should be noted that if any additional DH groups are specified, they must comply with the requirements (in terms of the ephemeral keys that are established) listed in FCS_CKM.1.

FCS_IPSEC_EXT.1.12 The TSF shall be able to ensure by default that the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [selection: *IKEv1 Phase 1, IKEv2 IKE_SA*] connection is greater than or equal to the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [selection: *IKEv1 Phase 2, IKEv2 CHILD_SA*] connection.

Application Note 66

The ST author chooses either or both of the IKE selections based on what is implemented by the TOE. Obviously, the IKE version(s) chosen should be consistent not only in this element, but with other choices for other elements in this component. While it is acceptable for this *capability to be configurable, the default configuration in the evaluated configuration (either "out of the box" or by configuration guidance in the AGD documentation) must enable this functionality.*

FCS_IPSEC_EXT.1.13 The TSF shall ensure that all IKE protocols perform peer authentication using a [selection: *RSA*, *ECDSA*] that use X.509v3 certificates that conform to RFC 4945 and [selection: *Pre-shared Keys, no other method*].

Application Note 67

At least one public-key-based Peer Authentication method is required in order to conform to this PP; one or more of the public key schemes is chosen by the ST author to reflect what is implemented. The ST author also ensures that appropriate FCS requirements reflecting the algorithms used (and key generation capabilities, if provided) are listed to support those methods. Note that the TSS will elaborate on the way in which these algorithms are to be used (for example, 2409 specifies three authentication methods using public keys; each one supported will be described in the TSS). Peer authentication using ECDSA X.509v3 certificates will be required for TOEs entering evaluation after Quarter 3, 2015.

FCS_IPSEC_EXT.1.14 The TSF shall only establish a trusted channel to peers with valid certificates.

Application Note 68

Supported peer certificate algorithms are the same as FCS_IPSEC_EXT.1.1.

(selection-based) **FCS_IPSEC_EXT.1.15** The TSF shall ensure that IKEv1 Phase 1 exchanges use only main mode.

Application Note 69

FCS_IPSEC_EXT.1.15 is only applicable if IKEv1 is selected in FCS_IPSEC_EXT.1.5.

B.2.1.3 FCS_SSHC_EXT.1 SSH Client Protocol

FCS_SSHC_EXT.1	**SSH Client Protocol**

FCS_SSHC_EXT.1.1 The TSF shall implement the SSH protocol that complies with RFCs 4251, 4252, 4253, 4254, and [selection: *5647, 5656, 6187, 6668, no other RFCs*].

Application Note 70

The ST author selects which of the additional RFCs to which conformance is being claimed. Note that these need to be consistent with selections in later elements of this component (e.g., cryptographic algorithms permitted). RFC 4253 indicates that certain cryptographic algorithms are "REQUIRED". This means that the implementation must include support, not that the algorithms must be enabled for use. Ensuring that algorithms indicated as "REQUIRED" but not listed in the later elements of this component are implemented is out of scope of the assurance activity for this requirement.

FCS_SSHC_EXT.1.2 The TSF shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based, password-based.

FCS_SSHC_EXT.1.3 The TSF shall ensure that, as described in RFC 4253, packets greater than [assignment: *number of bytes*] bytes in an SSH transport connection are dropped.

Application Note 71

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

FCS_SSHC_EXT.1.4 The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: *aes128-cbc, aes256-cbc, [selection: AEAD_AES_128_GCM, AEAD_AES_256_GCM, no other algorithms]*.

Application Note 72

RFC 5647 specifies the use of the AEAD_AES_128_GCM and AEAD_AES_256_GCM algorithms in SSH. As described in RFC 5647, AEAD_AES_128_GCM and AEAD_AES_256_GCM can only be chosen as encryption algorithms when the same algorithm is being used as the MAC algorithm. In the assignment, the ST author can select the AES-GCM algorithms, or "no other algorithms" if AES-GCM is not supported. If AES-GCM is selected, there should be corresponding FCS_COP entries in the ST.

FCS_SSHC_EXT.1.5 The TSF shall ensure that the SSH transport implementation uses *[selection: ssh-rsa, ecdsa-sha2-nistp256] and [selection: ecdsa-sha2-nistp384, x509v3-ecdsa-sha2-nistp256, x509v3-ecdsa-sha2-nistp384, no other public key algorithms]* as its public key algorithm(s) and rejects all other public key algorithms.

Application Note 73

Implementations that select only ssh-rsa will not achieve the 112-bit security strength in the digital signature generation for SSH authentication as is recommended in NIST SP 800-131A. Future versions of this profile may remove ssh-rsa as a selection. If x509v3-ecdsa-sha2-nistp256 or x509v3-ecdsa-sha2-nistp384 are selected, then the list of trusted certification authorities must be selected in FCS_SSHC_EXT.1.9.

FCS_SSHC_EXT.1.6 The TSF shall ensure that the SSH transport implementation uses *[selection: hmac-sha1, hmac-sha1-96, hmac-sha2-256, hmac-sha2-512] and [selection: AEAD_AES_128_GCM, AEAD_AES_256_GCM, no other MAC algorithms]* as its data integrity MAC algorithm(s) and rejects all other MAC algorithm(s).

Application Note 74

RFC 5647 specifies the use of the AEAD_AES_128_GCM and AEAD_AES_256_GCM algorithms in SSH. As described in RFC 5647, AEAD_AES_128_GCM and AEAD_AES_256_GCM can only be chosen as MAC algorithms when the same algorithm is being used as the encryption algorithm. RFC 6668 specifies the use of the sha2 algorithms in SSH.

FCS_SSHC_EXT.1.7 The TSF shall ensure that *[selection: diffie-hellman-group14-sha1, ecdh-sha2-nistp256] and [selection: ecdh-sha2-nistp384, ecdh-sha2-nistp521, no other methods]* are the only allowed key exchange methods used for the SSH protocol.

FCS_SSHC_EXT.1.8 The TSF shall ensure that the SSH connection be rekeyed after no more than 2^28 packets have been transmitted using that key.

FCS_SSHC_EXT.1.9 The TSF shall ensure that the SSH client authenticates the identity of the SSH server using a local database associating each host name with its corresponding public key or [selection: *a list of trusted certification authorities, no other methods*] as described in RFC 4251 section 4.1.

Application Note 75

The list of trusted certification authorities can only be selected if x509v3-ecdsa-sha2-nistp256 or x509v3-ecdsa-sha2-nistp384 are selected in FCS_SSHC_EXT.1.5.

B.2.1.4 FCS_SSHS_EXT.1 SSH Server Protocol

FCS_SSHS_EXT.1	SSH Server Protocol

FCS_SSHS_EXT.1.1 The TSF shall implement the SSH protocol that complies with RFCs 4251, 4252, 4253, 4254, and [selection: *5647, 5656, 6187, 6668, no other RFCs*].

Application Note 76

The ST author selects which of the additional RFCs to which conformance is being claimed. Note that these need to be consistent with selections in later elements of this component (e.g., cryptographic algorithms permitted). RFC 4253 indicates that certain cryptographic algorithms are "REQUIRED". This means that the implementation must include support, not that the algorithms must be enabled for use. Ensuring that algorithms indicated as "REQUIRED" but not listed in the later elements of this component are implemented is out of scope of the assurance activity for this requirement.

FCS_SSHS_EXT.1.2 The TSF shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based, password-based.

FCS_SSHS_EXT.1.3 The TSF shall ensure that, as described in RFC 4253, packets greater than [assignment: *number of bytes*] bytes in an SSH transport connection are dropped.

Application Note 77

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

FCS_SSHS_EXT.1.4 The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: *aes128-cbc, aes256-cbc, [selection: AEAD_AES_128_GCM, AEAD_AES_256_GCM, no other algorithms]*.

Application Note 78

RFC 5647 specifies the use of the AEAD_AES_128_GCM and AEAD_AES_256_GCM algorithms in SSH. As described in RFC 5647, AEAD_AES_128_GCM and AEAD_AES_256_GCM can only be chosen as encryption algorithms when the same algorithm is being used as the MAC algorithm. In the assignment, the ST author can select the AES-GCM algorithms, or "no other algorithms" if AES-GCM is not supported. If AES-GCM is selected, there should be corresponding FCS_COP entries in the ST.

FCS_SSHS_EXT.1.5 The TSF shall ensure that the SSH transport implementation uses *[selection: ssh-rsa, ecdsa-sha2-nistp256] and [selection: ecdsa-sha2-nistp384, x509v3-ecdsa-sha2-nistp256, x509v3-ecdsa-sha2-nistp384, no other public key algorithms]* as its public key algorithm(s) and rejects all other public key algorithms.

Application Note 79

Implementations that select only ssh-rsa will not achieve the 112-bit security strength in the digital signature generation for SSH authentication as is recommended in NIST SP 800-131A. Future versions of this profile may remove ssh-rsa as a selection.

FCS_SSHS_EXT.1.6 The TSF shall ensure that the SSH transport implementation uses *[selection: hmac-sha1, hmac-sha1-96, hmac-sha2-256, hmac-sha2-512] and [selection: AEAD_AES_128_GCM, AEAD_AES_256_GCM, no other MAC algorithms]* as its MAC algorithm(s) and rejects all other MAC algorithm(s).

Application Note 80

RFC 5647 specifies the use of the AEAD_AES_128_GCM and AEAD_AES_256_GCM algorithms in SSH. As described in RFC 5647, AEAD_AES_128_GCM and AEAD_AES_256_GCM can only be chosen as MAC algorithms when the same algorithm is being used as the encryption algorithm. RFC 6668 specifies the use of the sha2 algorithms in SSH.

FCS_SSHS_EXT.1.7 The TSF shall ensure that *[selection: diffie-hellman-group14-sha1, ecdh-sha2-nistp256] and [selection: ecdh-sha2-nistp384, ecdh-sha2-nistp521, no other methods]* are the only allowed key exchange methods used for the SSH protocol.

FCS_SSHS_EXT.1.8 The TSF shall ensure that the SSH connection be rekeyed after no more than 2^28 packets have been transmitted using that key.

B.2.1.5 FCS_TLSC_EXT.1 TLS Client Protocol

FCS_TLSC_EXT.1	**TLS Client Protocol**

FCS_TLSC_EXT.1.1 The TSF shall implement [selection: *TLS 1.2 (RFC 5246), TLS 1.1 (RFC 4346)*] supporting the following ciphersuites:

- *Mandatory Ciphersuites:*
  - *TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268*
- *[selection: Optional Ciphersuites:*
  - *TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268*

- *TLS_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268*
- *TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268*
- *TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492*
- *TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492*
- *TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492*
- *TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492*
- *TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246*
- *TLS_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246*
- *TLS_DHE_RSA_WITH_AES_128_CBC_ SHA256 as defined in RFC 5246*
- *TLS_DHE_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246*
- *TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289*
- *TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289*
- *TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289*
- *TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289*
- *no other ciphersuite].*

Application Note 81

The ciphersuites to be tested in the evaluated configuration are limited by this requirement. The ST author should select the optional ciphersuites that are supported; if there are no ciphersuites supported other than the mandatory suites, then "None" should be selected. It is necessary to limit the ciphersuites that can be used in an evaluated configuration administratively on the server in the test environment. The Suite B algorithms listed above (RFC 6460) are the preferred algorithms for implementation. TLS_RSA_WITH_AES_128_CBC_SHA is required in order to ensure compliance with RFC 5246.

These requirements will be revisited as new TLS versions are standardized by the IETF.

If any ciphersuites are selected using ECDHE, then FCS_TLSC_EXT.1.5 is required.

In a future version of this cPP TLS v1.2 will be required for all TOEs.

FCS_TLSC_EXT.1.2 The TSF shall verify that the presented identifier matches the reference identifier according to RFC 6125.

Application Note 82

The rules for verification of identify are described in Section 6 of RFC 6125. The reference identifier is established by the user (e.g. entering a URL into a web browser or clicking a link), by configuration (e.g. configuring the name of a mail server or authentication server), or by an application (e.g. a parameter of an API) depending on the application service. Based on a singular reference identifier's source domain and application service type (e.g. HTTP, SIP, LDAP), the client establishes all reference identifiers which are acceptable, such as a Common Name for the Subject Name field of the certificate and a (case-insensitive) DNS name, URI name, and Service Name for the Subject Alternative Name field. The client then compares this list of all acceptable reference identifiers to the presented identifiers in the TLS server's certificate.

The preferred method for verification is the Subject Alternative Name using DNS names, URI names, or Service Names. Verification using the Common Name is required for the purposes of backwards compatibility. Additionally, support for use of IP addresses in the Subject Name *or Subject Alternative name is discouraged as against best practices but may be implemented. Finally, the client should avoid constructing reference identifiers using wildcards. However, if the presented identifiers include wildcards, the client must follow the best practices regarding matching; these best practices are captured in the assurance activity.*

FCS_TLSC_EXT.1.3 The TSF shall only establish a trusted channel if the peer certificate is valid.

Application Note 83

Validity is determined by the identifier verification, certificate path, the expiration date, and the revocation status in accordance with RFC 5280. Certificate validity shall be tested in accordance with testing performed for FIA_X509_EXT.1.

FCS_TLSC_EXT.1.4 The TSF shall support mutual authentication using X.509v3 certificates.

Application Note 84

If TLS is used for FPT_ITC.1, then this component is required.

The use of X.509v3 certificates for TLS is addressed in FIA_X509_EXT.2.1. This requirement adds that this use must include the client must be capable of presenting a certificate to a TLS server for TLS mutual authentication.

FCS_TLSC_EXT.1.5 The TSF shall present the Supported Elliptic Curves Extension in the Client Hello with the following NIST curves: *[selection: secp256r1, secp384r1, secp521r1]* and no other curves.

Application Note 85

If ciphersuites with elliptic curves were selected in FCS_TLSC_EXT.1.1, this component is required.

This requirement limits the elliptic curves allowed for authentication and key agreement to the NIST curves from FCS_COP.1(2) and FCS_CKM.1 and FCS_CKM.2. This extension is required for clients supporting Elliptic Curve ciphersuites.

B.2.1.6 FCS_TLSS_EXT.1 TLS Server Protocol

FCS_TLSS_EXT.1	**TLS Server Protocol**

FCS_TLSS_EXT.1.1 The TSF shall implement [selection: *TLS 1.2 (RFC 5246), TLS 1.1 (RFC 4346)*] supporting the following ciphersuites:

- *Mandatory Ciphersuites:*
    - *TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268*
- *[selection: Optional Ciphersuites:*
    - *TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268*
    - *TLS_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268*
    - *TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268*

- *TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492*
- *TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492*
- *TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492*
- *TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492*
- *TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246*
- *TLS_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246*
- *TLS_DHE_RSA_WITH_AES_128_CBC_ SHA256 as defined in RFC 5246*
- *TLS_DHE_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246*
- *TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289*
- *TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289*
- *TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289*
- *TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289*
- *no other ciphersuite]*.

Application Note 86

The ciphersuites to be tested in the evaluated configuration are limited by this requirement. The ST author should select the optional ciphersuites that are supported; if there are no ciphersuites supported other than the mandatory suites, then "None" should be selected. It is necessary to limit the ciphersuites that can be used in an evaluated configuration administratively on the server in the test environment. The Suite B algorithms listed above (RFC 6460) are the preferred algorithms for implementation. TLS_RSA_WITH_AES_128_CBC_SHA is required in order to ensure compliance with RFC 5246.

These requirements will be revisited as new TLS versions are standardized by the IETF.

If any ciphersuites are selected using ECDHE, then FCS_TLSS_EXT.1.5 is required.

In a future version of this cPP TLS v1.2 will be required for all TOEs.

FCS_TLSS_EXT.1.2 The TSF shall deny connections from clients requesting SSL 1.0, SSL 2.0, SSL 3.0, TLS 1.0, and [selection: *TLS 1.1, none*].

Application Note 87

All SSL versions and TLS v1.0 shall be denied. Any TLS versions not selected in FCS_TLSS_EXT.1.1 should be selected here.

FCS_TLSS_EXT.1.3 The TSF shall generate key agreement parameters *[selection: over NIST curves [selection: secp256r1, secp384r1] and no other curves; Diffie-Hellman parameters of size 2048 bits and [selection: 3072 bits, no other size]]*.

Application Note 88

If the ST lists a DHE ciphersuite in FCS_TLSS_EXT.1.1, the ST must include the Diffie-Hellman selection in the requirement. FMT_SMF.1 requires the configuration of the key agreement parameters in order to establish the security strength of the TLS connection.

(optional) FCS_TLSS_EXT.1.4 The TSF shall support mutual authentication of TLS clients using X.509v3 certificates.

FCS_TLSS_EXT.1.5 The TSF shall not establish a trusted channel if the peer certificate is invalid.

Application Note 89

The use of X.509v3 certificates for TLS is addressed in FIA_X509_EXT.2.1. This requirement adds that this use must include support for client-side certificates for TLS mutual authentication.

Validity is determined by the certificate path, the expiration date, and the revocation status in accordance with RFC 5280. Certificate validity shall be tested in accordance with testing performed for FIA_X509_EXT.1.

FCS_TLSS_EXT.1.6 The TSF shall not establish a trusted channel if the distinguished name (DN) or Subject Alternative Name (SAN) contained in a certificate does not match the expected identifier for the peer.

Application Note 90

This requirement only applies to those TOEs performing mutually-authenticated TLS (FCS_TLSS_EXT.1.4). The peer identifier may be in the Subject field or the Subject Alternative Name extension of the certificate. The expected identifier may either be configured, may be compared to the Domain Name, IP address, username, or email address used by the peer, or may be passed to a directory server for comparison. Matching should be performed by a bit-wise comparison.

B.3 Protection of the TSF (FPT)

B.3.1 TSF self test (Extended)

B.3.1.1 FPT_TST_EXT.2 Self tests based on certificates

FPT_TST_EXT.2	Self tests based on certificates

FPT_TST_EXT.2.1 The TSF shall fail self-testing if a certificate is used for self tests and the corresponding certificate is deemed invalid.

Application Note 91

Certificates may optionally be used for self-tests (FPT_TST_EXT.1.1). This element must be included in the ST if certificates are used for self-tests. If "code signing for integrity verification" is selected in FIA_X509_EXT.2.1, FPT_TST_EXT.2.1 must be included in the ST.

Validity is determined by the certificate path, the expiration date, and the revocation status in accordance with FIA_X509_EXT.1.

B.3.2 Trusted Update (FPT_TUD_EXT)

B.3.2.1 FPT_TUD_EXT.2 Trusted Update based on certificates

FPT_TUD_EXT.2	Trusted Update based on certificates

FPT_TUD_EXT.2.1 The TSF shall not install an update if the code signing certificate is deemed invalid.

Application Note 92

Certificates may optionally be used for code signing of system software updates (FPT_TUD_EXT.1.3). This element must be included in the ST if certificates are used for validating updates. If "code signing for system software updates" is selected in FIA_X509_EXT.2.1, FPT_TUD_EXT.2.1 must be included in the ST.

Validity is determined by the certificate path, the expiration date, and the revocation status in accordance with FIA_X509_EXT.1.

B.4 Security Management (FMT)

B.4.1 Management of TSF Data (FMT_MTD)

B.4.1.1 FMT_MOF.1(1)/TrustedUpdate Management of TSF Data

FMT_MOF.1(2)/TrustedUpdate	Management of TSF Data

FMT_MOF.1.1(2)/TrustedUpdate The TSF shall restrict the ability to enable, disable the functions *automatic update* to *Security Administrators*.

Application Note 93

FMT_MOF.1(2)/TrustedUpdate is only applicable if the TOE supports automatic updates and allows to enable and disable them. Enable and disable of automatic updates is restricted to Security Administrators.

C. Extended Component Definitions

This appendix contains the definitions for the extended requirements that are used in the cPP, including those used in Appendices A and B.

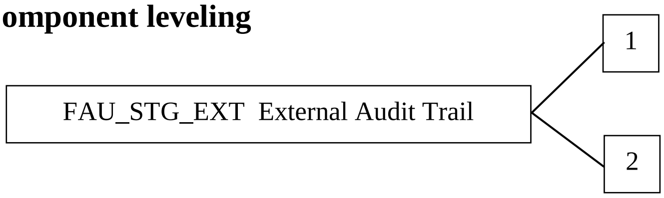
C.1 Security Audit (FAU)

C.1.1 Security audit event storage (FAU_STG_EXT)

Family Behaviour

This component defines the requirements for the TSF to be able to securely transmit audit data between the TOE and an external IT entity.

Component leveling



FAU_STG_EXT.1 External audit trail storage requires the TSF to use a trusted channel implementing a secure protocol.

FAU_STG_EXT.1 Counting lost audit data requires the TSF to provide information about audit records affected when the audit log becomes full.

Management: FAU_STG_EXT.1, FAU_STG_EXT.2

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

a) The TSF shall have the ability to configure the cryptographic functionality.

Audit: FAU_STG_EXT.1, FAU_STG_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) No audit necessary.

C.1.1.1 FAU_ STG_EXT.1 External Audit Trail Storage

FAU_STG_EXT.1	**Protected Audit Trail Storage**

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation
FTP_ITC.1 Inter-TSF Trusted Channel

FAU_STG_EXT.1.1 The TSF shall be able to [selection: *transmit the generated audit data to an external IT entity, receive and store audit data from an external IT entity*] using a trusted channel implementing the [selection: *IPsec, SSH, TLS, TLS/HTTPS*] protocol.

FAU_STG_EXT.2	**Counting lost audit data**

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation
FAU_STG_EXT.1 External Audit Trail Storage

FAU_STG_EXT.2.1 The TSF shall provide information about the number of [selection: *dropped, overwritten, assignment: other information*] audit records in the case where the local storage has been filled and the TSF takes one of the actions defined in FAU_STG_EXT.1.3.

C.2 Cryptographic Support (FCS)

C.2.1 Random Bit Generation (FCS_RBG_EXT)

C.2.1.1 FCS_RBG_EXT.1 Random Bit Generation

Family Behaviour

Components in this family address the requirements for random bit/number generation. This is a new family define do for the FCS class.

Component leveling

FCS_RBG_EXT.1 Extended: Random Bit Generation requires random bit generation to be performed in accordance with selected standards and seeded by an entropy source.

Management: FCS_RBG_EXT.1

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

a) There are no management activities foreseen

Audit: FCS_RBG_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: failure of the randomization process

FCS_RBG_EXT.1	**Random Bit Generation**

Hierarchical to: No other components

Dependencies: No other components

FCS_RBG_EXT.1.1 The TSF shall perform all deterministic random bit generation services in accordance with ISO/IEC 18031:2011 using [selection*: Hash_DRBG (any), HMAC_DRBG (any), CTR_DRBG (AES)]*.

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded by an entropy source that accumulates entropy from [selection: *a software-based noise source, a hardware-based noise source*] with minimum of [selection; *128 bits, 192 bits, 256 bits*] of entropy at least equal to the greatest security strength according to ISO/IEC 18031:2011 Table C.1 "Security Strength Table for Hash Functions" of the keys and hashes that it will generate.

Application Note 94

ISO/IEC 18031:2011contains three different methods of generating random numbers; each of these, in turn, depends on underlying cryptographic primitives (hash functions/ciphers). The ST author will select the function used, and include the specific underlying cryptographic primitives used in the requirement. While any of the identified hash functions (SHA-1, SHA-224, SHA-256, SHA-384, SHA-512) are allowed for Hash_DRBG or HMAC_DRBG, only AES-based implementations for CTR_DRBG are allowed.

C.2.2 Cryptographic Protocols (Extended – FCS_HTTPS_EXT, FCS_ IPSEC_EXT, FCS_SSHC_EXT, FCS_SSHS_EXT, FCS_TLSC_EXT, FCS_TLSS_EXT)

C.2.2.1 FCS_HTTPS_EXT.1 HTTPS Protocol

Family Behaviour

Components in this family define the requirements for protecting remote management sessions between the TOE and an Security Administrator. This family describes how HTTPS will be implemented. This is a new family defined for the FCS Class.

Component leveling

FCS_HTTPS_EXT HTTPS Protocol — 1

FCS_HTTPS_EXT.1 HTTPS requires that HTTPS be implemented according to RFC 2818 and supports TLS.

Management: FCS_HTTPS_EXT.1

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

a) There are no management activities foreseen.

Audit: FCS_HTTPS_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) There are no auditable events foreseen.

FCS_HTTPS_EXT.1 HTTPS Protocol

Hierarchical to: No other components

Dependencies: FCS_TLS_EXT.1 TLS Protocol

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

FCS_HTTPS_EXT.1.2 The TSF shall implement the HTTPS protocol using TLS.

FCS_HTTPS_EXT.1.3 The TSF shall [selection: *not establish the connection, request authorization to establish the connection, [assignment: other action]*] if the peer certificate is deemed invalid.

C.2.2.2 FCS_IPSEC_EXT.1 IPsec Protocol

Family Behaviour

Components in this family address the requirements for protecting communications using IPsec.

Component leveling

FCS_IPSEC_EXT.1 IPsec requires that IPsec be implemented as specified.

Management: FCS_IPSEC_EXT.1

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

a) Maintenance of SA lifetime configuration

Audit: FCS_IPSEC_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Decisions to DISCARD, BYPASS, PROTECT network packets processed by the TOE.
b) Failure to establish an IPsec SA

c) IPsec SA establishment
d) IPsec SA termination
e) Negotiation “down” from an IKEv2 to IKEv1 exchange.

FCS_IPSEC_EXT.1	**Internet Protocol Security (IPsec) Communications**

Hierarchical to: No other components

Dependencies: FCS_CKM.1 Cryptographic Key Generation
FCS_CKM.2 Cryptographic Key Establishment
FCS_COP.1(1) Cryptographic operation (AES Data encryption/decryption)
FCS_COP.1(2) Cryptographic operation (Signature Verification)
FCS_COP.1(3) Cryptographic Operation (Hash Algorithm)
FCS_RBG_EXT.1 Random Bit Generation

FCS_IPSEC_EXT.1.1 The TSF shall implement the IPsec architecture as specified in RFC 4301.

Application Note 95

RFC 4301 calls for an IPsec implementation to protect IP traffic through the use of a Security Policy Database (SPD). The SPD is used to define how IP packets are to be handled: PROTECT the packet (e.g., encrypt the packet), BYPASS the IPsec services (e.g., no encryption), or DISCARD the packet (e.g., drop the packet). The SPD can be implemented in various ways, including router access control lists, firewall rulesets, a “traditional” SPD, etc. Regardless of the implementation details, there is a notion of a “rule” that a packet is “matched” against and a resulting action that takes place.

While there must be a means to order the rules, a general approach to ordering is not mandated, as long as the SPD can distinguish the IP packets and apply the rules accordingly. There may be multiple SPDs (one for each network interface), but this is not required.

FCS_IPSEC_EXT.1.2 The TSF shall have a nominal, final entry in the SPD that matches anything that is otherwise unmatched, and discards it.

FCS_IPSEC_EXT.1.3 The TSF shall implement [selection: *tunnel mode*, *transport mode*].

FCS_IPSEC_EXT.1.4 The TSF shall implement the IPsec protocol ESP as defined by RFC 4303 using the cryptographic algorithms AES-CBC-128, AES-CBC-256 (both specified by RFC 3602) and [selection: *AES-GCM-128 (specified in RFC 4106)*, *AES-GCM-256 (specified in RFC 4106), no other algorithms*] together with a Secure Hash Algorithm (SHA)-based HMAC.

FCS_IPSEC_EXT.1.5 The TSF shall implement the protocol: [selection:

- *IKEv1 as defined in RFCs 2407, 2408, 2409, RFC 4109, [selection: no other RFCs for extended sequence numbers, RFC 4304 for extended sequence numbers], and [selection: no other RFCs for hash functions, RFC 4868 for hash functions];*

- *IKEv2 as defined in RFCs 5996 [selection: with no support for NAT traversal, with mandatory support for NAT traversal as specified in RFC 5996, section 2.23)], and [selection: no other RFCs for hash functions, RFC 4868 for hash functions]*].

FCS_IPSEC_EXT.1.6 The TSF shall ensure the encrypted payload in the [selection: *IKEv1, IKEv2*] protocol uses the cryptographic algorithms AES-CBC-128, AES-CBC-256 as specified in RFC 3602 and [selection: *AES-GCM-128, AES-GCM-256 as specified in RFC 5282, no other algorithm*].

Application Note 96

AES-GCM-128 and AES-GCM-256 may only be selected if IKEv2 is also selected, as there is no RFC defining AES-GCM for IKEv1.

FCS_IPSEC_EXT.1.7 The TSF shall ensure that [selection:

- *IKEv1 Phase 1 SA lifetimes can be configured by a Security Administrator based on [selection:*
  - *number of packets/number of bytes;*
  - *length of time, where the time values can configured within [assignment: integer range including 24] hours;*

  ];
- *IKEv2 SA lifetimes can be configured by a Security Administrator based on [selection:*
  - *number of packets/number of bytes;*
  - *length of time, where the time values can configured within [assignment: integer range including 24] hours*

  ]

].

Application Note 97

The ST author chooses either the IKEv1 requirements or IKEv2 requirements (or both, depending on the selection in FCS_IPSEC_EXT.1.5). The ST author chooses either packet/volume-based lifetimes or time-based lifetimes. This requirement must be accomplished by providing Security Administrator-configurable lifetimes. Hardcoded limits do not meet this requirement.

FCS_IPSEC_EXT.1.8 The TSF shall ensure that [selection:

- *IKEv1 Phase 2 SA lifetimes can be configured by a Security Administrator based on [selection:*
  - *number of packets/number of bytes;*
  - *length of time, where the time values can be configured within [assignment: integer range including 8] hours;*

  ];

- *IKEv2 Child SA lifetimes can be configured by a Security Administrator based on [selection:*
    - *number of packets/number of bytes;*
    - *length of time, where the time values can be configured within [assignment: integer range including 8] hours;*

    ]

].

Application Note 98

The ST author chooses either the IKEv1 requirements or IKEv2 requirements (or both, depending on the selection in FCS_IPSEC_EXT.1.5). The ST author chooses either packet/volume-based lifetimes or time-based lifetimes. This requirement must be accomplished by providing Security Administrator-configurable lifetimes. Hardcoded limits do not meet this requirement.

FCS_IPSEC_EXT.1.9 The TSF shall generate the secret value x used in the IKE Diffie-Hellman key exchange ("x" in g^x mod p) using the random bit generator specified in FCS_RBG_EXT.1, and having a length of at least [assignment: *(one or more) number(s) of bits that is at least twice the security strength of the negotiated Diffie-Hellman group*] bits.

Application Note 99

For DH groups 19 and 20, the "x" value is the point multiplier for the generator point G.

Since the implementation may allow different Diffie-Hellman groups to be negotiated for use in forming the SAs, the assignment in FCS_IPSEC_EXT.1.11 may contain multiple values. For each DH group supported, the ST author consults Table 2 in NIST SP 800-57 "Recommendation for Key Management –Part 1: General" to determine the security strength ("bits of security") associated with the DH group. Each unique value is then used to fill in the assignment for this element. For example, suppose the implementation supports DH group 14 (2048-bit MODP) and group 20 (ECDH using NIST curve P-384). From Table 2, the bits of security value for group 14 is 112, and for group 20 it is 192.

FCS_IPSEC_EXT.1.10 The TSF shall generate nonces used in [selection: *IKEv1*, *IKEv2*] exchanges of length [selection:

- *[assignment: security strength associated with the negotiated Diffie-Hellman group];*
- *at least 128 bits in size and at least half the output size of the negotiated pseudorandom function (PRF) hash*

  ] .

Application Note 100

The ST author must select the second option for nonce lengths if IKEv2 is also selected (as this is mandated in RFC 5996). The ST author may select either option for IKEv1.

For the first option for nonce lengths, since the implementation may allow different Diffie-Hellman groups to be negotiated for use in forming the SAs, the assignment in FCS_IPSEC_EXT.1.11 may contain multiple values. For each DH group supported, the ST author consults Table 2 in NIST SP 800-57 "Recommendation for Key Management –Part 1: General" to determine the security strength ("bits of security") associated with the DH *group. Each unique value is then used to fill in the assignment for this element. For example, suppose the implementation supports DH group 14 (2048-bit MODP) and group 20 (ECDH using NIST curve P-384). From Table 2, the bits of security value for group 14 is 112, and for group 20 it is 192.*

Because nonces may be exchanged before the DH group is negotiated, the nonce used should be large enough to support all TOE-chosen proposals in the exchange.

FCS_IPSEC_EXT.1.11 The TSF shall ensure that all IKE protocols implement DH Groups 14 (2048-bit MODP), and [selection: *19 (256-bit Random ECP), 5 (1536-bit MODP), 24 (2048-bit MODP with 256-bit POS), 20 (384-bit Random ECP), [assignment: other DH groups that are implemented by the TOE], no other DH groups*].

FCS_IPSEC_EXT.1.12 The TSF shall be able to ensure by default that the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [selection: *IKEv1 Phase 1, IKEv2 IKE_SA*] connection is greater than or equal to the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [selection: *IKEv1 Phase 2, IKEv2 CHILD_SA*] connection.

Application Note 101

The ST author chooses either or both of the IKE selections based on what is implemented by the TOE. While it is acceptable for this capability to be configurable, the default configuration in the evaluated configuration (either "out of the box" or by configuration guidance in the AGD documentation) must enable this functionality.

FCS_IPSEC_EXT.1.13 The TSF shall ensure that all IKE protocols perform peer authentication using a [selection: *RSA, ECDSA*] that use X.509v3 certificates that conform to RFC 4945 and [selection: *Pre-shared Keys, no other method*].

FCS_IPSEC_EXT.1.14 The TSF shall only establish a trusted channel to peers with valid certificates.

FCS_IPSEC_EXT.1.15 The TSF shall ensure that IKEv1 Phase 1 exchanges use only main mode.

C.2.2.3 FCS_SSHC_EXT.1 SSH Client

Family Behaviour

The component in this family addresses the ability for a client to use SSH to protect data between the client and a server using the SSH protocol.

Component leveling

FCS_SSHC_EXT.1 Extended: SSH Client requires that the client side of SSH be implemented as specified.

Management: FCS_SSHC_EXT.1

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

a) There are no management activities foreseen.

Audit: FCS_SSHC_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Failure of SSH session establishment.
b) SSH session establishment
c) SSH session termination

FCS_SSHC_EXT.1	**SSH Client Protocol**

Hierarchical to: No other components

Dependencies: FCS_COP.1(1) Cryptographic operation (AES Data encryption/decryption)
FCS_COP.1(2) Cryptographic operation (Signature Verification)
FCS_COP.1(3) Cryptographic Operation (Hash Algorithm)

FCS_SSHC_EXT.1.1 The TSF shall implement the SSH protocol that complies with RFCs 4251, 4252, 4253, 4254, and [selection: *5647, 5656, 6187, 6668, no other RFCs*].

Application Note 102

The ST author selects which of the additional RFCs to which conformance is being claimed. Note that these need to be consistent with selections in later elements of this component (e.g., cryptographic algorithms permitted). RFC 4253 indicates that certain cryptographic algorithms are "REQUIRED". This means that the implementation must include support, not that the algorithms must be enabled for use. Ensuring that algorithms indicated as "REQUIRED" but not listed in the later elements of this component are implemented is out of scope of the assurance activity for this requirement.

FCS_SSHC_EXT.1.2 The TSF shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based, password-based.

FCS_SSHC_EXT.1.3 The TSF shall ensure that, as described in RFC 4253, packets greater than [assignment: *number of bytes*] bytes in an SSH transport connection are dropped.

Application Note 103

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

FCS_SSHC_EXT.1.4 The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: [assignment: *List of encryption algorithms*].

FCS_SSHC_EXT.1.5 The TSF shall ensure that the SSH transport implementation uses [assignment: *List of public key algorithms*] as its public key algorithm(s) and rejects all other public key algorithms.

FCS_SSHC_EXT.1.6 The TSF shall ensure that the SSH transport implementation uses [assignment: *List of data integrity MAC algorithms*] as its data integrity MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SSHC_EXT.1.7 The TSF shall ensure that [assignment: *List of key exchange methods*] are the only allowed key exchange methods used for the SSH protocol.

FCS_SSHC_EXT.1.8 The TSF shall ensure that the SSH connection be rekeyed after no more than 2^28 packets have been transmitted using that key.

FCS_SSHC_EXT.1.9 The TSF shall ensure that the SSH client authenticates the identity of the SSH server using a local database associating each host name with its corresponding public key or [selection: *a list of trusted certification authorities, no other methods*] as described in RFC 4251 section 4.1.

Application Note 104

The list of trusted certification authorities can only be selected if x509v3-ecdsa-sha2-nistp256 or x509v3-ecdsa-sha2-nistp384 are specified in FCS_SSHC_EXT.1.5.

C.2.2.4 FCS_SSHS_EXT.1 SSH Server Protocol

Family Behaviour

The component in this family addresses the ability for a server to offer SSH to protect data between a client and the server using the SSH protocol.

Component leveling

FCS_SSHS_EXT.1 Extended: SSH Server requires that the server side of SSH be implemented as specified.

Management: FCS_SSHS_EXT.1

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

a) There are no management activities foreseen.

Audit: FCS_SSHS_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Failure of SSH session establishment.
b) SSH session establishment
c) SSH session termination

FCS_SSHS_EXT.1	**SSH Server Protocol**

Hierarchical to: No other components

Dependencies: FCS_COP.1(1) Cryptographic operation (AES Data encryption/decryption)
FCS_COP.1(2) Cryptographic operation (Signature Verification)
FCS_COP.1(3) Cryptographic Operation (Hash Algorithm)

FCS_SSHS_EXT.1.1 The TSF shall implement the SSH protocol that complies with RFCs 4251, 4252, 4253, 4254, and [selection: *5647, 5656, 6187, 6668, no other RFCs*].

Application Note 105

The ST author selects which of the additional RFCs to which conformance is being claimed. Note that these need to be consistent with selections in later elements of this component (e.g., cryptographic algorithms permitted). RFC 4253 indicates that certain cryptographic algorithms are "REQUIRED". This means that the implementation must include support, not that the algorithms must be enabled for use. Ensuring that algorithms indicated as "REQUIRED" but not listed in the later elements of this component are implemented is out of scope of the assurance activity for this requirement.

FCS_SSHS_EXT.1.2 The TSF shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based, password-based.

FCS_SSHS_EXT.1.3 The TSF shall ensure that, as described in RFC 4253, packets greater than [assignment: *number of bytes*] bytes in an SSH transport connection are dropped.

Application Note 106

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

FCS_SSHS_EXT.1.4 The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: [assignment: *encryption algorithms*].

FCS_SSHS_EXT.1.5 The TSF shall ensure that the SSH transport implementation uses [assignment: *List of public key algorithms*] as its public key algorithm(s) and rejects all other public key algorithms.

FCS_SSHS_EXT.1.6 The TSF shall ensure that the SSH transport implementation uses [assignment: *List of MAC algorithms*] as its MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SSHS_EXT.1.7 The TSF shall ensure that [assignment: *List of key exchange methods*] are the only allowed key exchange methods used for the SSH protocol.

FCS_SSHS_EXT.1.8 The TSF shall ensure that the SSH connection be rekeyed after no more than 2^28 packets have been transmitted using that key.

C.2.2.5 FCS_TLSC_EXT.1 TLS Client Protocol

Family Behaviour

The component in this family addresses the ability for a client to use TLS to protect data between the client and a server using the TLS protocol.

Component leveling

FCS_TLSC_EXT.1 TLS Client requires that the client side of TLS be implemented as specified.

Management: FCS_TLSC_EXT.1

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

a) There are no management activities foreseen.

Audit: FCS_TLSC_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Failure of TLS session establishment.
b) TLS session establishment
c) TLS session termination

FCS_TLSC_EXT.1	**TLS Client Protocol**

Hierarchical to: No other components

Dependencies: FCS_COP.1(1) Cryptographic operation (AES Data encryption/decryption)
FCS_COP.1(2) Cryptographic operation (Signature Verification)

FCS_COP.1(3) Cryptographic Operation (Hash Algorithm)
FCS_RBG_EXT.1 Random Bit Generation

FCS_TLSC_EXT.1.1 The TSF shall implement [selection: *TLS 1.2 (RFC 5246), TLS 1.1 (RFC 4346)*] supporting the following ciphersuites:

- *Mandatory Ciphersuites:*
    - *[assignment: List of mandatory ciphersuites and reference to RFC in which each is defined]*
- *[selection: Optional Ciphersuites:*
    - *[assignment: List of optional ciphersuites and reference to RFC in which each is defined]*
    - *no other ciphersuite]]*.

Application Note 107

The ciphersuites to be tested in the evaluated configuration are limited by this requirement. Note that TLS_RSA_WITH_AES_128_CBC_SHA is required in order to ensure compliance with RFC 5246. If any ciphersuites using ECDHE are specified, then FCS_TLSC_EXT.1.5 is required.

FCS_TLSC_EXT.1.2 The TSF shall verify that the presented identifier matches the reference identifier according to RFC 6125.

Application Note 108

The rules for verification of identify are described in Section 6 of RFC 6125. The reference identifier is established by the user (e.g. entering a URL into a web browser or clicking a link), by configuration (e.g. configuring the name of a mail server or authentication server), or by an application (e.g. a parameter of an API) depending on the application service. Based on a singular reference identifier's source domain and application service type (e.g. HTTP, SIP, LDAP), the client establishes all reference identifiers which are acceptable, such as a Common Name for the Subject Name field of the certificate and a (case-insensitive) DNS name, URI name, and Service Name for the Subject Alternative Name field. The client then compares this list of all acceptable reference identifiers to the presented identifiers in the TLS server's certificate.

FCS_TLSC_EXT.1.3 The TSF shall only establish a trusted channel if the peer certificate is valid.

Application Note 109

Validity is determined by the identifier verification, certificate path, the expiration date, and the revocation status in accordance with RFC 5280.

FCS_TLSC_EXT.1.4 The TSF shall support mutual authentication using X.509v3 certificates.

Application Note 110

The use of X.509v3 certificates for TLS is addressed in FIA_X509_EXT.2.1. This requirement adds that this use must include the client must be capable of presenting a certificate to a TLS server for TLS mutual authentication.

FCS_TLSC_EXT.1.5 The TSF shall present the Supported Elliptic Curves Extension in the Client Hello with the following NIST curves: [assignment: *List of supported curves*].

Application Note 111

If ciphersuites with elliptic curves were specified in FCS_TLSC_EXT.1.1, this component is required.

C.2.2.6 FCS_TLSS_EXT.1 TLS Server Protocol

Family Behaviour

The component in this family addresses the ability for a server to use TLS to protect data between a client and the server using the TLS protocol.

Component leveling

FCS_TLSS_EXT TLS Server Protocol	1

FCS_TLSS_EXT.1 Extended: TLS Server requires that the server side of TLS be implemented as specified.

Management: FCS_TLSS_EXT.1

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

a) There are no management activities foreseen.

Audit: FCS_TLSS_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Failure of TLS session establishment.
b) TLS session establishment
c) TLS session termination

FCS_TLSS_EXT.1	**TLS Server Protocol**

Hierarchical to:	No other components
Dependencies:	FCS_CKM.1 Cryptographic Key Generation FCS_COP.1(1) Cryptographic operation (AES Data encryption/decryption) FCS_COP.1(2) Cryptographic operation (Signature Verification) FCS_COP.1(3) Cryptographic Operation (Hash Algorithm) FCS_RBG_EXT.1 Random Bit Generation

FCS_TLSS_EXT.1.1 The TSF shall implement [selection: *TLS 1.2 (RFC 5246)*, *TLS 1.1 (RFC 4346)*] supporting the following ciphersuites:

- *Mandatory Ciphersuites:*
  - *[assignment: List of mandatory ciphersuites and reference to RFC in which each is defined]*
- *[selection: Optional Ciphersuites:*
  - *[assignment: List of optional ciphersuites and reference to RFC in which each is defined]*
  - *no other ciphersuite]]*.

Application Note 112

The ciphersuites to be tested in the evaluated configuration are limited by this requirement. Note that TLS_RSA_WITH_AES_128_CBC_SHA is required in order to ensure compliance with RFC 5246. If any ciphersuites using ECDHE are specified, then FCS_TLSC_EXT.1.5 is required.

FCS_TLSS_EXT.1.2 The TSF shall deny connections from clients requesting SSL 1.0, SSL 2.0, SSL 3.0, TLS 1.0, and [selection: *TLS 1.1, none*].

Application Note 113

Any TLS versions not selected in FCS_TLSS_EXT.1.1 should be selected here.

FCS_TLSS_EXT.1.3 The TSF shall generate key agreement parameters [selection: [assignment: *List of elliptic curves*]; [assignment: *List of diffie-hellman parameter sizes*]].

Application Note 114

The assignments will be filled in based on the assignments performed in FCS_TLSS_EXT.1.1.

FCS_TLSS_EXT.1.4 The TSF shall support mutual authentication of TLS clients using X.509v3 certificates.

Application Note 115

The use of X.509v3 certificates for TLS is addressed in FIA_X509_EXT.2.1. This requirement adds that this use must include support for client-side certificates for TLS mutual authentication.

FCS_TLSS_EXT.1.5 The TSF shall not establish a trusted channel if the peer certificate is invalid.

Application Note 116

Validity is determined by the certificate path, the expiration date, and the revocation status in accordance with RFC 5280.

FCS_TLSS_EXT.1.6 The TSF shall not establish a trusted channel if the distinguished name (DN) or Subject Alternative Name (SAN) contained in a certificate does not match the expected identifier for the peer.

Application Note 117

This requirement only applies to those TOEs performing mutually-authenticated TLS (FCS_TLSS_EXT.1.4). The peer identifier may be in the Subject field or the Subject Alternative Name extension of the certificate. The expected identifier may either be configured, may be compared to the Domain Name, IP address, username, or email address used by the peer, or may be passed to a directory server for comparison.

C.3 Firewall (FFW)

C.3.1 Stateful Traffic Filter Firewall (FFW_RUL_EXT)

Family Behaviour

This requirement is used to specify the behavior of a Stateful Traffic Filter Firewall. The network protocols that the TOE can filter, as well as the attributes that can be used by an administrator to construct a ruleset are identified in this component. How the ruleset is processed (i.e., ordering) is specified, as well as any expected default behavior on the part of the TOE.

Component leveling

FFW_RUL_EXT.1 Stateful Traffic Filtering requires the TOE to filter network traffic based on a ruleset configured by an authorized administrator.

Management: FFW_RUL_EXT.1

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

a) enable/disable a ruleset on a network interface
b) configure a ruleset
c) specifying rules that govern the use of resources

Audit: FFW_RUL_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal:
   - Result (i.e., drop, allow) of applying a rule in the ruleset to a network packet
   - Configuration of the ruleset
   - Indication of packets dropped due to too much network traffic

C.3.1.1 FFW_RULEXT.1 Stateful Traffic Filtering

FFW_RULEXT.1	Stateful Traffic Filtering

Hierarchical to: No other components

Dependencies: None

FFW_RUL_EXT.1.1 The TSF shall perform Stateful Traffic Filtering on network packets processed by the TOE.

FFW_RUL_EXT.1.2 The TSF shall allow the definition of Stateful Traffic Filtering rules using the following network protocol fields: [assignment: *list of attributes supported by the ruleset*].

FFW_RUL_EXT.1.3 The TSF shall allow the following operations to be associated with Stateful Traffic Filtering rules: permit or drop with the capability to log the operation.

FFW_RUL_EXT.1.4 The TSF shall allow the Stateful Traffic Filtering rules to be assigned to each distinct network interface.

FFW_RUL_EXT.1.5 The TSF shall:

1. accept a network packet without further processing of Stateful Traffic Filtering rules if it matches an allowed established session for the following protocols: [assignment: *list of supported protocols for which state is maintained*] based on the following network packet attributes: [assignment: *list of attributes associated with each of the protocols*].
2. Remove existing traffic flows from the set of established traffic flows based on the following: [selection: *session inactivity timeout, completion of the expected information flow*].

[OPTIONAL] FFW_RUL_EXT.1.6 The TSF shall dynamically define rules or establish sessions allowing network traffic to flow for the following network protocols [assignment: *list of supported protocols*].

FFW_RUL_EXT.1.7 The TSF shall enforce the following default Stateful Traffic Filtering rules on all network traffic: [assignment: *list of default rules that are applied to network traffic flow*].

FFW_RUL_EXT.1.8 The TSF shall be capable of dropping and logging according to the following rules: [assignment: *list of specific rules that the TOE is capable of enforcing*]

FFW_RUL_EXT.1.9 The TSF shall process the applicable Stateful Traffic Filtering rules in an administratively defined order.

FFW_RUL_EXT.1.10 The TSF shall deny packet flow if a matching rule is not identified.

FFW_RUL_EXT.1.11 The TSF shall be capable of limiting an administratively configured number of [assignment: *rules governing the use of resources*].

C.4 Identification and Authentication (FIA)

C.4.1 Password Management (FIA_PMG_EXT)

Family Behaviour

The TOE defines the attributes of passwords used by administrative users to ensure that strong passwords and passphrases can be chosen and maintained.

Component leveling

FIA_PMG_EXT.1 Password management requires the TSF to support passwords with varying composition requirements, minimum lengths, maximum lifetime, and similarity constraints.

Management: FIA_PMG_EXT.1

No management functions.

Audit: FIA_PMG_EXT.1

No specific audit requirements.

C.4.1.1 FIA_PMG_EXT.1 Password Management

FIA_PMG_EXT.1	**Password Management**
Hierarchical to:	No other components.
Dependencies:	No other components.

FIA_PMG_EXT.1.1 The TSF shall provide the following password management capabilities for administrative passwords:

1. *Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: [selection: “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”, [assignment: other characters]];*
2. *Minimum password length shall settable by the Security Administrator, and support passwords of 15 characters or greater.*

C.4.2 User Identification and Authentication (FIA_UIA_EXT)

Family Behaviour

The TSF allows certain specified actions before the non-TOE entity goes through the identification and authentication process.

Component leveling

FIA_UIA_EXT User Identification and Authentication — 1

FIA_UIA_EXT.1 User Identification and Authentication requires administrators (including remote administrators) to be identified and authenticated by the TOE, providing assurance for that end of the communication path. It also ensures that every user is identified and authenticated before the TOE performs any mediated functions

Management: FIA_UIA_EXT.1

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

a) Ability to configure the list of TOE services available before an entity is identified and authenticated

Audit: FIA_UIA_EXT.N

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) All use of the identification and authentication mechanism
b) Provided user identity, origin of the attempt (e.g. IP address)

C.4.2.1 FIA_UIA_EXT.1 User Identification and Authentication

FIA_UIA_EXT.1	**User Identification and Authentication**

Hierarchical to: No other components.

Dependencies: FTA_TAB.1 Default TOE Access Banners

FIA_UIA_EXT.1.1 The TSF shall allow the following actions prior to requiring the non-TOE entity to initiate the identification and authentication process:

- Display the warning banner in accordance with FTA_TAB.1;
- [selection: *no other actions, [assignment: list of services, actions performed by the TSF in response to non-TOE requests.]*]

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

C.4.3 User authentication (FIA_UAU) (FIA_UAU_EXT)

Family Behaviour

Provides for a locally based administrative user authentication mechanism

Component leveling

FIA_UAU_EXT.1 The password-based authentication mechanism provides administrative users a locally based authentication mechanism..

Management: FIA_UAU_EXT.2

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

a) None

Audit: FIA_UAU_EXT.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: All use of the authentication mechanism

C.4.3.1 FIA_UAU_EXT.2 Password-based Authentication Mechanism

FIA_UAU_EXT.2	Password-based Authentication Mechanism

Hierarchical to: No other components.

Dependencies: None

FIA_UAU_EXT.2.1 The TSF shall provide a local password-based authentication mechanism, [selection: *[assignment: other authentication mechanism(s)]*, *none*] to perform administrative user authentication.

C.4.4 Authentication using X.509 certificates (Extended – FIA_X509_EXT)

Family Behaviour

This family defines the behavior, management, and use of X.509 certificates for functions to be performed by the TSF. Components in this family require validation of certificates according to a specified set of rules, use of certificates for authentication for protocols and integrity verification, and the generation of certificate requests.

Component leveling

FIA_X509_EXT.1 X509 Certificate Validation, requires the TSF to check and validate certificates in accordance with the RFCs and rules specified in the component.

FIA_X509_EXT.2 X509 Certificate Authentication, requires the TSF to use certificates to authenticate peers in protocols that support certificates, as well as for integrity verification and potentially other functions that require certificates.

FIA_X509_EXT.3 X509 Certificate Requests, requires the TSF to be able to generate Certificate Request Messages and validate responses.

Management: FIA_X509_EXT.1, FIA_X509_EXT.2, FIA_X509_EXT.3

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

a) Remove imported X.509v3 certificates
b) Approve import and removal of X.509v3 certificates
c) Initiate certificate requests

Audit: FIA_X509_EXT.1, FIA_X509_EXT.2, FIA_X509_EXT.3

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: No specific audit requirements are specified.

C.4.4.1 FIA_X509_EXT.1 X.509 Certificate Validation

FIA_X509_EXT.1	X.509 Certificate Validation

Hierarchical to: No other components

Dependencies: No other components

FIA_X509_EXT.1.1 The TSF shall validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certificate path validation.
- The certificate path must terminate with a trusted certificate.

- The TSF shall validate a certificate path by ensuring the presence of the basicConstraints extension and that the CA flag is set to TRUE for all CA certificates.

- The TSF shall validate the revocation status of the certificate using [selection: *the Online Certificate Status Protocol (OCSP) as specified in RFC 2560, a Certificate Revocation List (CRL) as specified in RFC 5759*].

- The TSF shall validate the extendedKeyUsage field according to the following rules: [assignment: *rules that govern contents of the extendedKeyUsage field that need to be verified*].

Application Note 118

FIA_X509_EXT.1.1 lists the rules for validating certificates. The ST author shall select whether revocation status is verified using OCSP or CRLs. The ST author fills in the assignment with rules that may apply to other requirements in the ST. For instance, if a protocol such as TLS that uses certificates is specified in the ST, then certain values for the extendedKeyUsage field (e.g., "Server Authentication Purpose") could be specified.

FIA_X509_EXT.1.2 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 119

This requirement applies to certificates that are used and processed by the TSF and restricts the certificates that may be added as trusted CA certificates.

C.4.4.2 FIA_X509_EXT.2 X509 Certificate Authentication

FIA_X509_EXT.2	**X.509 Certificate Authentication**

Hierarchical to: No other components

Dependencies: No other components

FIA_X509_EXT.2.1 The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for [selection*: IPsec, TLS, HTTPS, SSH, [assignment: other protocols], no protocols*], and [selection: *code signing for system software updates, code signing for integrity verification, [assignment: other uses], no additional uses*].

Application Note 120

If the TOE specifies the implementation of communications protocols that perform peer authentication using certificates, the ST author either selects or assigns the protocols that are specified; otherwise, they select "no protocols". The TOE may also use certificates for other purposes; the second selection and assignment are used to specify these cases.

FIA_X509_EXT.2.2 When the TSF cannot establish a connection to determine the validity of a certificate, the TSF shall [selection: *allow the administrator to choose whether to accept the certificate in these cases, accept the certificate, not accept the certificate*].

Application Note 121

Often a connection must be established to check the revocation status of a certificate - either to download a CRL or to perform a lookup using OCSP. The selection is used to describe the behavior in the event that such a connection cannot be established (for example, due to a network error). If the TOE has determined the certificate valid according to all other rules in FIA_X509_EXT.1, the behavior indicated in the selection shall determine the validity.

C.4.4.3 FIA_X509_EXT.3 X.509 Certificate Requests

FIA_X509_EXT.3	**X.509 Certificate Requests**

Hierarchical to: No other components

Dependencies: No other components

FIA_X509_EXT.3.1 The TSF shall generate a Certificate Request Message as specified by RFC 2986 and be able to provide the following information in the request: public key and [selection: *device-specific information, Common Name, Organization, Organizational Unit, Country, [assignment: other information]*].

FIA_X509_EXT.3.2 The TSF shall validate the chain of certificates from the Root CA upon receiving the CA Certificate Response.

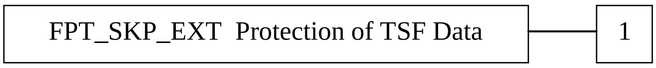
C.5 Protection of the TSF (FPT)

C.5.1 Protection of TSF Data (FPT_SKP_EXT)

Family Behaviour

Components in this family address the requirements for managing and protecting TSF data, such as cryptographic keys. This is a new family modelled after the FPT_PTD Class.

Component leveling



FPT_SKP_EXT.1 Protection of TSF Data (for reading all symmetric keys), requires preventing symmetric keys from being read by any user or subject. It is the only component of this family.

Management: FPT_SKP_EXT.1

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

a) There are no management activities foreseen.

Audit: FPT_SKP_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) There are no auditable events foreseen.

C.5.1.1 FPT_SKP_EXT.1 Protection of TSF Data (for reading of all symmetric keys)

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

Hierarchical to: No other components.

Dependencies: No other components.

FPT_SKP_EXT.1.1 The TSF shall prevent reading of all pre-shared keys, symmetric keys, and private keys.

C.5.2 Protection of Administrator Passwords (FPT_APW_EXT)

C.5.2.1 FPT_APW_EXT.1 Protection of Administrator Passwords

Family Behaviour

Components in this family ensure that the TSF will protect plaintext credential data such as passwords from unauthorized disclosure.

Component leveling

FPT_APW_EXT.1 Protection of administrator passwords requires that the TSF prevent plaintext credential data from being read by any user or subject.

Management: FPT_APW_EXT.1

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

a) No management functions.

Audit: FPT_APW_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) No audit necessary.

FPT_APW_EXT.1	**Protection of Administrator Passwords**

Hierarchical to: No other components

Dependencies: No other components.

FPT_APW_EXT.1.1 The TSF shall store passwords in non-plaintext form.

FPT_APW_EXT.1.2 The TSF shall prevent the reading of plaintext passwords.

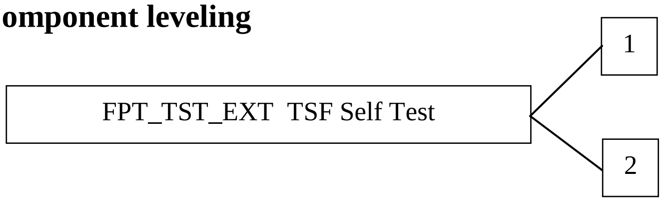
C.5.3 TSF self test

C.5.3.1 FPT_TST_EXT.1 TSF Testing

Family Behaviour

Components in this family address the requirements for self-testing the TSF for selected correct operation.

Component leveling



FPT_TST_EXT.1 TSF Self Test requires a suite of self tests to be run during initial start-up in order to demonstrate correct operation of the TSF.

FPT_TST_EXT.2 Self tests based on certificates applies when using certificates as part of self test, and requires that the self test fails if a certificate is invalid.

Management: FPT_TST_EXT.1, FPT_TST_EXT.2

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

a) No management functions.

Audit: FPT_TST_EXT.1, FPT_TST_EXT.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Indication that TSF self test was completed

FPT_TST_EXT.1 TSF testing

Hierarchical to: No other components.

Dependencies: None

FPT_TST_EXT.1.1 The TSF shall run a suite of the following self-tests [selection: *during initial start-up (on power on), periodically during normal operation, at the request of the authorised user, at the conditions [assignment: conditions under which self-tests should occur]*] to demonstrate the correct operation of the TSF: [assignment: *list of self-tests run by the TSF during initial start-up*].

FPT_TST_EXT.2 Self tests based on certificates

Hierarchical to: No other components.

Dependencies: None

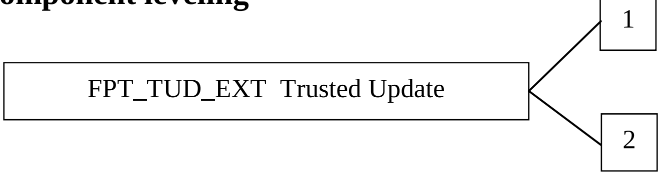
FPT_TST_EXT.2.1 The TSF shall fail self-testing if a certificate is used for self tests and the corresponding certificate is deemed invalid.

C.5.4 Trusted Update (FPT_TUD_EXT)

Family Behaviour

Components in this family address the requirements for updating the TOE firmware and/or software.

Component leveling



FPT_TUD_EXT.1 Trusted Update requires management tools be provided to update the TOE firmware and software, including the ability to verify the updates prior to installation.

FPT_TUD_EXT.2 Trusted update based on certificates applies when using certificates as part of trusted update, and requires that the update does not install if a certificate is invalid.

Management: FPT_TUD_EXT.1

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

a) Ability to update the TOE and to verify the updates
b) Ability to update the TOE and to verify the updates using the digital signature capability (FCS_COP.1(2)) and [selection: no other functions, [assignment: other cryptographic functions (or other functions) used to support the update capability]]
c) Ability to update the TOE, and to verify the updates using [selection: digital signature, published hash, no other mechanism] capability prior to installing those updates

Audit: FPT_TUD_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Initiation of the update process.
b) Any failure to verify the integrity of the update

C.5.4.1 FPT_TUD_EXT.1 Trusted Update

FPT_TUD_EXT.1	**Trusted update**

Hierarchical to: No other components

Dependencies: FCS_COP.1(1) Cryptographic operation (for cryptographic signature), or FCS_COP.1(3) Cryptographic operation (for cryptographic hashing)

FPT_TUD_EXT.1.1 The TSF shall provide [selection: *[assignment: role or group]*, *none*] the ability to query the currently executed version of the TOE firmware/software as well as the most recently installed version of the TOE firmware/software.

FPT_TUD_EXT.1.2 The TSF shall provide [selection: *[assignment: role or group]*, *none*] the ability to manually initiate updates to TOE firmware/software and [selection: *support automatic updates*, *no other update mechanism*].

FPT_TUD_EXT.1.3 The TSF shall provide means to authenticate firmware/software updates to the TOE using a digital signature mechanism prior to installing those updates.

C.5.4.2 FPT_TUD_EXT.2 Trusted Update based on certificates

FPT_TUD_EXT.2	**Trusted update based on certificates**

Hierarchical to: No other components

Dependencies: FPT_TUD_EXT.1

FPT_TUD_EXT.2.1 The TSF shall not install an update if the code signing certificate is deemed invalid.

C.6 TOE Access (FTA)

C.6.1 FTA_SSL_EXT.1 TSF-initiated Session Locking

Family Behaviour

Components in this family address the requirements for TSF-initiated and user-initiated locking, unlocking, and termination of interactive sessions.

The extended FTA_SSL_EXT family is based on the FTA_SSL family.

Component leveling

FTA_SSL_EXT.1 Extended: TSF-initiated session locking, requires system initiated locking of an interactive session after a specified period of inactivity. It is the only component of this family.

Management: FTA_SSL_EXT.1

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

c) Specification of the time of user inactivity after which lock-out occurs for an individual user.

Audit: FTA_SSL_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Any attempts at unlocking an interactive session.

FTA_SSL_EXT.1	**TSF-initiated Session Locking**

Hierarchical to: No other components

Dependencies: FIA_UAU.1 Timing of authentication

FTA_SSL_EXT.1.1 The TSF shall, for local interactive sessions, [selection:

- *lock the session - disable any activity of the user's data access/display devices other than unlocking the session, and requiring that the administrator re-authenticate to the TSF prior to unlocking the session;*
- *terminate the session*]

after a Security Administrator-specified time period of inactivity.

D. Entropy Documentation And Assessment

This appendix describes the required supplementary information for the entropy source used by the TOE.

The documentation of the entropy source should be detailed enough that, after reading, the evaluator will thoroughly understand the entropy source and why it can be relied upon to provide entropy. This documentation should include multiple detailed sections: design description, entropy justification, operating conditions, and health testing. This documentation is not required to be part of the TSS.

D.1 Design Description

Documentation shall include the design of the entropy source as a whole, including the interaction of all entropy source components. It will describe the operation of the entropy source to include how it works, how entropy is produced, and how unprocessed (raw) data can be obtained from within the entropy source for testing purposes. The documentation should walk through the entropy source design indicating where the random comes from, where it is passed next, any post-processing of the raw outputs (hash, XOR, etc.), if/where it is stored, and finally, how it is output from the entropy source. Any conditions placed on the process (e.g., blocking) should also be described in the entropy source design. Diagrams and examples are encouraged.

This design must also include a description of the content of the security boundary of the entropy source and a description of how the security boundary ensures that an adversary outside the boundary cannot affect the entropy rate.

If implemented, the design description shall include a description of how third-party applications can add entropy to the RBG. A description of any RBG state saving between power-off and power-on shall be included.

D.2 Entropy Justification

There should be a technical argument for where the unpredictability in the source comes from and why there is confidence in the entropy source exhibiting probabilistic behavior (an explanation of the probability distribution and justification for that distribution given the particular source is one way to describe this). This argument will include a description of the expected entropy rate and explain how you ensure that sufficient entropy is going into the TOE randomizer seeding process. This discussion will be part of a justification for why the entropy source can be relied upon to produce bits with entropy.

The entropy justification shall not include any data added from any third-party application or from any state saving between restarts.

D.3 Operating Conditions

Documentation will also include the range of operating conditions under which the entropy source is expected to generate random data. It will clearly describe the measures that have been taken in the system design to ensure the entropy source continues to operate under those conditions. Similarly, documentation shall describe the conditions under which the entropy source is known to malfunction or become inconsistent. Methods used to detect failure or degradation of the source shall be included.

D.4 Health Testing

More specifically, all entropy source health tests and their rationale will be documented. This will include a description of the health tests, the rate and conditions under which each health test is performed (e.g., at startup, continuously, or on-demand), the expected results for each health test, and rationale indicating why each test is believed to be appropriate for detecting one or more failures in the entropy source.

E. Glossary

Term	Meaning
Assurance	Grounds for confidence that a TOE meets the SFRs [CC1].
Key Chaining	The method of using multiple layers of encryption keys to protect data. A top layer key encrypts a lower layer key which encrypts the data; this method can have any number of layers.
Target of Evaluation	A set of software, firmware and/or hardware possibly accompanied by guidance. [CC1]
TOE Security Functionality (TSF)	A set consisting of all hardware, software, and firmware of the TOE that must be relied upon for the correct enforcement of the SFRs. [CC1]
TSF Data	Data for the operation of the TSF upon which the enforcement of the requirements relies.

See [CC1] for other Common Criteria abbreviations and terminology.

F. Acronyms

Acronym	Meaning
AEAD	Authenticated Encryption with Associated Data
AES	Advanced Encryption Standard
CA	Certificate Authority
CBC	Cipher Block Chaining
CCM	Counter with CBC-Message Authentication Code
CRL	Certificate Revocation List
DH	Diffie-Hellman
DSA	Digital Signature Algorithm
ECDH	Elliptic Curve Diffie Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
EEPROM	Electrically Erasable Programmable Read-Only Memory
FIPS	Federal Information Processing Standards
FQDN	Fully Qualified Domain Name
GCM	Galois Counter Mode
HMAC	Keyed-Hash Message Authentication Code
HTTPS	HyperText Transfer Protocol Secure
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
IPsec	Internet Protocol Security
NIST	National Institute of Standards and Technology
OCSP	Online Certificate Status Protocol
OS	Operating System
PP	Protection Profile
RA	Registration Authority
RBG	Random Bit Generator
ROM	Read-only memory
RSA	Rivest Shamir Adleman Algorithm
SD	Supporting Document
SHA	Secure Hash Algorithm
SPI	Security Parameter Index
SSH	Secure Shell
ST	Security Target
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Functionality
TSS	TOE Summary Specification
VPN	Virtual Private Network
XCCDF	eXtensible Configuration Checklist Description Format
XTS	XEX (XOR Encrypt XOR) Tweakable Block Cipher with Ciphertext Stealing