Extended Component Definitions

This appendix contains the definitions for the extended requirements used in the PP, including those used in Appendices B and C.

Background and Scope

This Appendix provides a definition for all the extended components introduced in this PP. These components are identified in the following table:

Functional Class	Functional Components
Cryptographic Support (FCS)	FCS_CKM_EXT Cryptographic Key Management
	FCS_RBG_EXT Random Bit Generation
	FCS_STO_EXT Storage of Sensitive Data
User Data Protection (FDP)	FDP_ACF_EXT Access Controls for Protecting User Data
	FDP_IFC_EXT Information Flow Control
Identification and Authentication (FIA)	FIA_X509_EXT Authentication Using X.509 Certificates
Security Management (FMT)	FMT_MOF_EXT Management of Functions Behavior
	FMT_SMF_EXT Specification of Management Functions
Protection of the TSF (FPT)	FPT_ACF_EXT Access Controls
	FPT_ASLR_EXT Address Space Layout Randomization
	FPT_BLT_EXT Limitation of Bluetooth Profile Support
	FPT_SBOP_EXT Stack Buffer Overflow Protection
	FPT_SRP_EXT Software Restriction Policies
	FPT_TST_EXT Boot Integrity
	FPT_TUD_EXT Trusted Update
	FPT_W^X_EXT Write XOR Execute Memory Pages
Trusted Path/Channels (FTP)	FTP_ITC_EXT Trusted Channel Communication

Extended Component Definitions

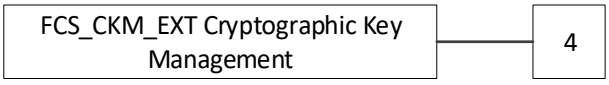
Class FCS: Cryptographic Support

FCS_CKM_EXT Cryptographic Key Management

Family Behavior

This family defines requirements for key management. It differs from FCS_CKM in CC Part 2 by defining technology-specific details for the implementation of these functions.

Component Leveling



FCS_CKM_EXT.4, Cryptographic Key Destruction, requires the TSF to destroy cryptographic keys based on one or more specific methods, depending on the physical medium on which the key data is stored. Note that 4 was chosen for the family's sole component number to show that this requirement is similar to FCS_CKM.4, from which it was originally derived.

Management: FCS_CKM_EXT.4

There are no management activities foreseen.

Audit: FCS_CKM_EXT.4

There are no auditable events foreseen.

FCS_CKM_EXT.4 Cryptographic Key Destruction

Hierarchical to: No other components

Dependencies: No dependencies

FCS_CKM_EXT.4.1 The OS shall destroy cryptographic keys and key material in accordance with a specified cryptographic key destruction method [**selection**:

- *For volatile memory, the destruction shall be executed by a [**selection**:*
    - *single overwrite consisting of [**selection**: a pseudo-random pattern using the TSF's DRBG, zeroes, ones, a new value of a key [**assignment**: any value that does not contain any CSP]]*
    - *removal of power to the memory*
    - *destruction of reference to the key direction followed by a request for garbage collection*

  ]

- *For non-volatile memory that consists of [**selection**:*
    - *destruction of all key encrypting keys (KEKs) protecting the target key according to FCS_CKM_EXT.4.1, where none of the KEKs protecting the target key are derived*
    - *the invocation of an interface provided by the underlying platform that [**selection**:*
        - *logically addresses the storage location of the key and performs a [**selection**: single, [**assignment**: ST author defined multi-pass] overwrite consisting of [**selection**: zeroes, ones, pseud-random pattern, a new value of a key of the same size, [**assignment**: any value that does not contain any CSP]]]*
        - *instructs the underlying platform to destroy the abstraction that represents the key*

    ]

  ]

].

FCS_CKM_EXT.4.2 The OS shall destroy all keys and key material when no longer needed.

FCS_RBG_EXT Random Bit Generation

Family Behavior

Components in this family address the requirements for random bit and number generation. This is a new family defined for the FCS class.

Component Leveling

FCS_RBG_EXT.1, 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

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:

- Failure of the randomization process

FCS_RBG_EXT.1 Random Bit Generation

Hierarchical to: No other components

Dependencies: FCS_COP.1 Cryptographic Operation

FCS_RBG_EXT.1.1 The OS shall perform all deterministic random bit generation (DRBG) services in accordance with NIST Special Publication 800-90A using [**selection**:

- *Hash_DRBG (any)*
- *HMAC_DRBG (any)*
- *CTR_DRBG (AES)*

].

FCS_RBG_EXT.1.2 The deterministic RBG used by the OS shall be seeded by an entropy source that accumulates entropy from a [**selection**:

- *software-based noise source*

- *platform-based noise source*

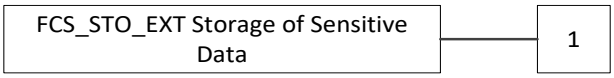
] with a minimum of 256 bits of entropy at least equal to the greatest security strength (according to NIST SP 800-57) of the keys and hashes that it will generate.

FCS_STO_EXT Storage of Sensitive Data

Family Behavior

Components in this family describe the requirements for storing sensitive data (such as cryptographic keys). This is a new family defined for the FCS class.

Component Leveling



FCS_STO_EXT.1, Storage of Sensitive Data, requires the TSF to include a mechanism that encrypts sensitive data and that can be invoked by third-party applications in addition to internal TSF usage.

Management: FCS_STO_EXT.1

There are no management activities foreseen.

Audit: FCS_STO_EXT.1

There are no auditable events foreseen.

FCS_STO_EXT.1 Storage of Sensitive Data

Hierarchical to: No other components

Dependencies: FCS_COP.1 Cryptographic Operation

FCS_STO_EXT.1.1 The OS shall implement functionality to encrypt sensitive data stored in non-volatile storage and provide interfaces to applications to invoke this functionality.

Class FDP: User Data Protection

FDP_ACF_EXT Access Controls for Protecting User Data

Family Behavior

This family specifies methods for ensuring that data stored or maintained by the TSF cannot be accessed without authorization. This family differs from FDP_ACF in CC Part 2 by defining technology-specific details for the implementation of these functions.

Component Leveling

FDP_ACF_EXT.1, Access Controls for Protecting User Data, requires the TSF to prevent unprivileged users from accessing operating system objects owned by other users.

Management: FDP_ACF_EXT.1

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

- Configuration of object ownership and allowed access

Audit: FDP_ACF_EXT.1

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

- Successful and unsuccessful attempts to access data

FDP_ACF_EXT.1 Access Controls for Protecting User Data

Hierarchical to: No other components

Dependencies: No dependencies

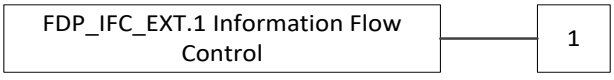
FDP_ACF_EXT.1.1 The OS shall implement access controls which can prohibit unprivileged users from accessing files and directories owned by other users.

FDP_IFC_EXT Information Flow Control

Family Behavior

This family defines the ability of the TSF to control information flows by ensuring that it is possible to use IPsec to encapsulate all traffic bound to or from the TOE. This family differs from FDP_IFC in CC Part 2 by defining technology-specific details for the implementation of these functions.

Component Leveling



FDP_IFC_EXT.1, Information Flow Control, requires the TSF to provide the ability to protect IP traffic using IPsec.

Management: FDP_IFC_EXT.1

There are no management activities foreseen.

Audit: FDP_IFC_EXT.1

There are no auditable events foreseen.

FDP_IFC_EXT.1 Information Flow Control

Hierarchical to: No other components

Dependencies: FTP_ITC_EXT.1 Trusted Channel Communication

FDP_IFC_EXT.1.1 The OS shall [**selection**:

- *provide an interface which allows a VPN client to protect all IP traffic using IPsec*
- *provide a VPN client which can protect all IP traffic using IPsec*

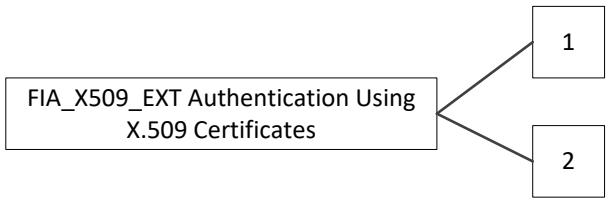
] with the exception of IP traffic required to establish the VPN connection and [**selection**: *signed updates directly from the OS vendor, no other traffic*].

Class FIA: Identification and Authentication

FIA_X509_EXT Authentication Using X.509 Certificates

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 and use of certificates for authentication for protocols and integrity verification. This is a new family defined for the FCS class.

Component Leveling



FIA_X509_EXT.1, X.509 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, X.509 Certificate Authentication, requires the TSF to use certificates for authentication functions.

Management: FIA_X509_EXT.1

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

- Import and removal of X.509v3 certificates
- Approval of import and removal of X.509v3 certificates

Audit: FIA_X509_EXT.1

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

- Failure of certificate validation

Management: FIA_X509_EXT.2

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

- Import and removal of X.509v3 certificates
- Approval of import and removal of X.509v3 certificates

Audit: FIA_X509_EXT.2

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

- Success or failure of authentication attempt

FIA_X509_EXT.1 X.509 Certificate Validation

Hierarchical to: No other components

Dependencies: FCS_COP.1 Cryptographic Operation

FPT_STM.1 Reliable Time Stamps

FIA_X509_EXT.1.1 The OS shall implement functionality to validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certificate path validation
- The certificate path must terminate with a trusted CA certificate
- The OS shall validate a certificate path by ensuring the presence of the basicConstraints extension, that the CA flag is set to TRUE for all CA certificates, and that any path constraints are met.
- The TSF shall validate that any CA certificate includes "Certificate Signing" as a purpose the key usage field
- The OS shall validate the revocation status of the certificate using [**selection**: *OCSP as specified in RFC 6960, CRL as specified in RFC 8603, an OCSP TLS Status Request Extension (OCSP stapling) as specified in RFC 6066, OCSP TLS Multi-Certificate Status Request Extension (i.e., OCSP Multi-Stapling) as specified in RFC 6961*] with [**selection**: *no exceptions, [**assignment**: exceptional use cases and alternative status check]*]
- The OS 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 EKU field.

- S/MIME certificates presented for email encryption and signature shall have the Email Protection purpose (id-kp 4 with OID 1.3.6.1.5.5.7.3.4) in the EKU 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 EKU field.
- Server certificates presented for EST shall have the CMC Registration Authority (RA) purpose (id-kp-cmcRA with OID 1.3.6.1.5.5.7.3.28) in the EKU field. (conditional)

.

FIA_X509_EXT.1.2 The OS shall only treat a certificate as a CA certificate if the basicConstraints extension is present and the CA flag is set to TRUE.

FIA_X509_EXT.2 X.509 Certificate Authentication

Hierarchical to: No other components

Dependencies: FIA_X509_EXT.1 Certificate Validation

FIA_X509_EXT.2.1 The OS shall use X.509v3 certificates as defined by RFC 5280 to support authentication for [**selection**: *TLS, DTLS, HTTPS, [**assignment**: other protocols]*] connections.

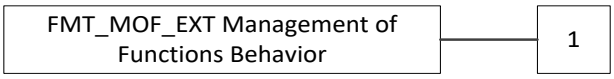
Class FMT: Security Management

FMT_MOF_EXT Management of Functions Behavior

Family Behavior

This family defines the administrative privileges required to modify the behavior of the security functions that are defined specifically for operating systems.

Component Leveling



FMT_MOF_EXT.1, Management of Security Functions Behavior, requires the TSF to define a set of management functions for the TOE and the privileges that are required to administer them.

Management: FMT_MOF_EXT.1

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

- Configuration of the roles that may manage the behavior of the TSF management functions

Audit: FMT_MOF_EXT.1

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

- Successful or unsuccessful management of the behavior of any TOE functions

- Change in permissions to set of users that have the ability to manage a given function

FMT_MOF_EXT.1 Management of Functions Behavior

Hierarchical to: No other components

Dependencies: FMT_SMF_EXT.1 Specification of Management Functions

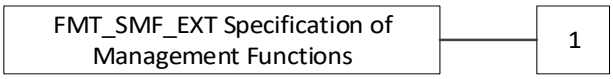
FMT_MOF_EXT.1.1 The OS shall restrict the ability to perform the function indicated in the "Administrator" column in FMT_SMF_EXT.1.1 to the administrator.

FMT_SMF_EXT Specification of Management Functions

Family Behavior

This family defines management functions that are defined specifically for operating systems.

Component Leveling



FMT_SMF_EXT.1, Specification of Management Functions, requires the TSF to define a set of management functions for the TOE.

Management: FMT_SMF_EXT.1

There are no management activities foreseen.

Audit: FMT_SMF_EXT.1

There are no auditable events foreseen.

FMT_SMF_EXT.1 Specification of Management Functions

Hierarchical to: No other components

Dependencies: No dependencies

FMT_SMF_EXT.1.1 The OS shall be capable of performing the following management functions:

#	Management Function	Administrator	User
1	Enable/disable [**selection**: *screen lock, session timeout*]	M	O
2	Configure [**selection**: *screen lock, session*] inactivity timeout	M	O
3	Import keys/secrets into the secure key storage	O	O
4	Configure local audit storage capacity	O	O
5	Configure minimum password length	O	O
6	Configure minimum number of special characters in password	O	O
7	Configure minimum number of numeric characters in password	O	O

#	Management Function	Administrator	User
8	Configure minimum number of uppercase characters in password	O	O
9	Configure minimum number of lowercase characters in password	O	O
10	Configure lockout policy for unsuccessful authentication attempts through [**selection**: *timeouts between attempts, limiting number of attempts during a time period*]	O	O
11	Configure host-based firewall	O	O
12	Configure name/address of directory server with which to bind	O	O
13	Configure name/address of remote management server from which to receive management settings	O	O
14	Configure name/address of audit/logging server to which to send audit/logging records	O	O
15	Configure audit rules	O	O
16	Configure name/address of network time server	O	O
17	Enable/disable automatic software update	O	O
18	Configure Wi-Fi interface	O	O
19	Enable/disable Bluetooth interface	O	O
20	Enable/disable [***assignment****: list of other external interfaces*]	O	O
21	[***assignment****: list of other management functions to be provided by the TSF*]	O	O

Class FPT: Protection of the TSF

FPT_ACF_EXT Access Controls

Family Behavior

This family defines specific TOE components that are protected against unprivileged access. This is a new family defined for the FCS class.

Component Leveling

FPT_ACF_EXT.1, Access Controls, requires the TSF to prohibit unauthorized users from reading or modifying specific TSF data.

Management: FPT_ACF_EXT.1

No specific management functions are identified.

Audit: FPT_ACF_EXT.1

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

- Unauthorized attempts to perform operations against protected data

FPT_ACF_EXT.1 Access Controls

Hierarchical to: No other components

Dependencies: No dependencies

FPT_ACF_EXT.1.1 The OS shall implement access controls which prohibit unprivileged users from modifying:

- Kernel and its drivers/modules
- Security audit logs
- Shared libraries
- System executables
- System configuration files
- [**assignment***: other objects*]

.

FPT_ACF_EXT.1.2 The OS shall implement access controls which prohibit unprivileged users from reading:

- Security audit logs
- System-wide credential repositories
- [**assignment***: list of other objects*]

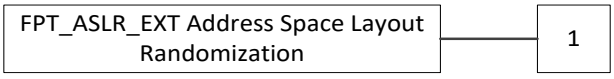
.

FPT_ASLR_EXT Address Space Layout Randomization

Family Behavior

This family defines the ability of the TOE to implement address space layout randomization (ASLR). This is a new family defined for the FCS class.

Component Leveling



FPT_ASLR_EXT.1, Address Space Layout Randomization, defines the ability of the TOE to use ASLR as well as the objects that ASLR is applied to.

Management: FPT_ASLR_EXT.1

There are no management functions foreseen.

Audit: FPT_ASLR_EXT.1

There are no auditable events foreseen.

FPT_ASLR_EXT.1 Address Space Layout Randomization

Hierarchical to: No other components

Dependencies: No dependencies

FPT_ASLR_EXT.1.1 The OS shall always randomize process address space memory locations with [**selection**: *8,* [***assignment****: number greater than 8]*] bits of entropy except for [**assignment***: list of explicit exceptions*].

FPT_BLT_EXT Limitation of Bluetooth Profile Support

Family Behavior

This family defines requirements for limiting Bluetooth capabilities without user action. This is a new family defined for the FCS class.

Component Leveling

FPT_BLT_EXT.1, Limitation of Bluetooth Profile Support, requires the TSF to maintain a disabled by default posture for Bluetooth profiles.

Management: FPT_BLT_EXT.1

There are no management activities foreseen.

Audit: FPT_BLT_EXT.1

There are no auditable events foreseen.

FPT_BLT_EXT.1 Limitation of Bluetooth Profile Support

Hierarchical to: No other components

Dependencies: No dependencies

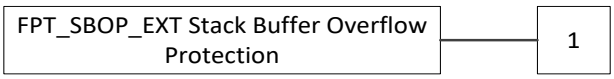
FPT_BLT_EXT.1 The TSF shall disable support for [**assignment**: list of Bluetooth profiles] Bluetooth profiles when they are not currently being used by an application on the TOE and shall require explicit user action to enable them.

FPT_SBOP_EXT Stack Buffer Overflow Protection

Family Behavior

This family requires the TSF to be compiled using stack-based buffer overflow protections. This is a new family defined for the FCS class.

Component Leveling



FPT_SBOP_EXT.1, Stack Buffer Overflow Protection, requires the TSF to be compiled using stack-based buffer overflow protections or to store data in such a manner that a stack-based buffer overflow cannot compromise the TSF.

Management: FPT_SBOP_EXT.1

There are no management functions foreseen.

Audit: FPT_SBOP_EXT.1

There are no auditable events foreseen.

FPT_SBOP_EXT.1 Stack Buffer Overflow Protection

Hierarchical to: No other components

Dependencies: No dependencies

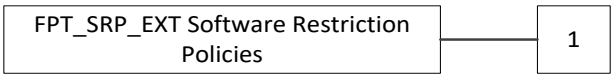
FPT_SBOP_EXT.1 The OS shall [**selection**: *employ stack-based buffer overflow protections, not store parameters/variables in the same data structures as control values*].

FPT_SRP_EXT Software Restriction Policies

Family Behavior

This family defines the ability of the TOE to restrict the execution of software unless it meets defined criteria. This is a new family defined for the FCS class.

Component Leveling



FPT_SRP_EXT.1, Software Restriction Policies, defines the criteria the TSF can use to prevent execution of restricted programs.

Management: FPT_SRP_EXT.1

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

- Specification of restriction policies

Audit: FPT_SRP_EXT.1

There are no auditable events foreseen.

FPT_SRP_EXT.1 Software Restriction Policies

Hierarchical to: No other components

Dependencies: No dependencies

FPT_SRP_EXT.1.1 The OS shall restrict execution to only programs which match an administrator-specified [**selection**:

- *file path*
- *file digital signature*
- *version*
- *hash*
- *[**assignment**: other characteristics]*

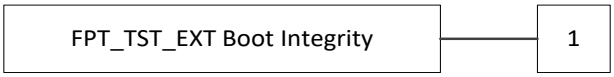
].

FPT_TST_EXT Boot Integrity

Family Behavior

This family defines the ability of the TOE to provide a mechanism that can be used to verify its integrity when started.

Component Leveling



FPT_TST_EXT.1, Boot Integrity, defines the mechanisms that the TSF uses to assert its own integrity at startup.

Management: FPT_TST_EXT.1

There are no management functions foreseen.

Audit: FPT_TST _EXT.1

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

- Failure of the integrity checking mechanism

FPT_TST _EXT.1 Boot Integrity

Hierarchical to: No other components

Dependencies: FCS_COP.1 Cryptographic Operation

FIA_X509_EXT.1 X.509 Certificate Validation

FPT_TST_EXT.1.1 The OS shall verify the integrity of the bootchain up through the OS kernel and [**selection**:

- *all executable code stored in mutable media*
- *[**assignment**: list of other executable code]*
- *no other executable code*

] prior to its execution through the use of [**selection**:

- *a digital signature using a hardware-protected asymmetric key*
- *a digital signature using an X509 certificate with hardware-based protection*
- *a hardware-protected hash*

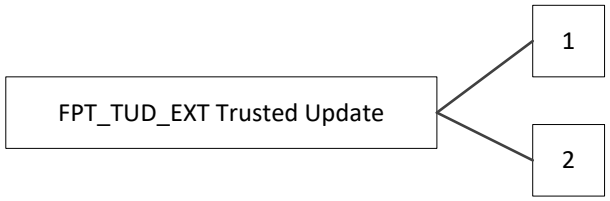
].

FPT_TUD_EXT Trusted Update

Family Behavior

This family defines the ability of the TOE to provide mechanisms for assuring the integrity of updates to the TSF or to non-TOE components that that rely on the TSF to function. This is a new family defined for the FCS class.

Component Leveling



FPT_TUD_EXT.1, Trusted Update, requires the TOE to provide a mechanism to verify the integrity of updates to itself.

FPT_TUD_EXT.2, Trusted Update for Application Software, requires the TOE to provide a mechanism to verify the integrity of updates to non-TSF applications that are running on the TOE.

Management: FPT_TUD_EXT.1

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

- Configuration of update checking mechanism
- Initiation of update

Audit: FPT_TUD _EXT.1

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

- Failure of the integrity checking mechanism
- Successful completion of updates

Management: FPT_TUD_EXT.2

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

- Configuration of update checking mechanism
- Initiation of update

Audit: FPT_TUD _EXT.2

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

- Failure of the integrity checking mechanism
- Successful completion of updates

FPT_TUD_EXT.1 Integrity for Installation and Update

Hierarchical to: No other components

Dependencies: FCS_COP.1 Cryptographic Operation

FPT_TUD_EXT.1.1 The OS shall provide the ability to check for updates to the OS software itself and shall use a digital signature scheme specified in FCS_COP.1/SIGN to validate the authenticity of the response.

FPT_TUD_EXT.1.2 The OS shall [**selection**: *cryptographically verify, invoke platform-provided functionality to cryptographically verify*] updates to itself using a digital signature prior to installation using schemes specified in FCS_COP.1/SIGN.

FPT_TUD_EXT.2 Integrity for Installation and Update of Application Software

Hierarchical to: No other components

Dependencies: FCS_COP.1 Cryptographic Operation

FPT_TUD_EXT.2.1 The OS shall provide the ability to check for updates to application software and shall use a digital signature scheme specified in FCS_COP.1 to validate the authenticity of the response.

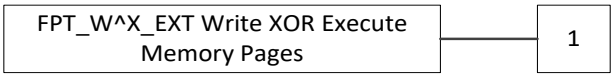
FPT_TUD_EXT.2.2 The OS shall cryptographically verify the integrity of updates to applications using a digital signature specified by FCS_COP.1 prior to installation.

FPT_W^X_EXT Write XOR Execute Memory Pages

Family Behavior

This family defines the ability of the TOE to implement data execution prevention (DEP) by preventing memory from being both writable and executable. This is a new family defined for the FCS class.

Component Leveling



FPT_W^X_EXT.1, Write XOR Execute Memory Pages, defines the ability of the TOE to prevent memory from being simultaneously writable and executable unless otherwise specified.

Management: FPT_W^X_EXT.1

There are no management functions foreseen.

Audit: FPT_W^X_EXT.1

There are no auditable events foreseen.

FPT_W^X_EXT.1 Write XOR Execute Memory Pages

Hierarchical to: No other components

Dependencies: No dependencies

FPT_W^X_EXT.1.1 The OS shall prevent allocation of any memory region with both write and execute permissions except for [**assignment***: list of exceptions*].

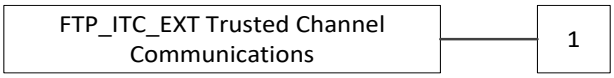
Class FTP: Trusted Path/Channels

FTP_ITC_EXT Trusted Channel Communication

Family Behavior

This family defines the ability of the TOE to use specific trusted communications channels to communicate with specific non-TOE entities in the Operational Environment. This family differs from FTP_ITC in Part 2 by defining technology-specific details for the implementation of these functions.

Component Leveling



FTP_ITC_EXT.1, Trusted channel communications, defines the specific secure communications protocols the TSF uses to communicate with a specific set of non-TOE entities in the Operational Environment.

Management: FTP_ITC_EXT.1

No specific management functions are identified.

Audit: FTP_ITC_EXT.1

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

- Initiation of trusted channel
- Termination of trusted channel
- Failure of trusted channel functions

FTP_ITC_EXT.1 Trusted Channel Communication

Hierarchical to: No other components

Dependencies: FCS_DTLS_EXT.1 DTLS Implementation
FCS_IPSEC_EXT.1 IPsec
FCS_SSH_EXT.1 SSH Protocol
FCS_TLSC_EXT.1 TLS Client Protocol

FTP_ITC_EXT.1.1 The OS shall use [**selection***:*

- *TLS as conforming to the Functional Package for Transport Layer Security (TLS), version 1.1 as a [**selection**: client, server]*
- *DTLS as conforming to the Functional Package for Transport Layer Security (TLS), version 1.1 as a [**selection**: client, server]*
- *IPsec as conforming to the PP-Module for Virtual Private Network (VPN) Clients, version 2.4*
- *SSH as conforming to the Functional Package for Secure Shell (SSH), version 1.0 as a [**selection**: client, server]*

] to provide a trusted communication channel between itself and authorized IT entities supporting the following capabilities: [**selection**: *audit server, authentication server, management 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.