Protection Profile for General-Purpose Computing Platforms

Version: 0.1

2019-12-13

National Information Assurance Partnership

Revision History

Version	Date	Comment
0.1	2019-12-13	Initial Draft

Contents

1 Introduction

1.1 Overview

The scope of this Protection Profile (PP) is to describe the security functionality of general-purpose computing platforms in terms of [CC] and to define security functional and assurance requirements for such products.

1.2 Terms

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

1.2.1 Common Criteria Terms

Assurance	Grounds for confidence that a TOE meets the SFRs [CC].
Common Criteria (CC)	Common Criteria for Information Technology Security Evaluation.
Common Evaluation Methodology (CEM)	Common Evaluation Methodology for Information Technology Security Evaluation.
Distributed TOE	A TOE composed of multiple components operating as a logical whole.
Operational Environment	Hardware and software that are outside the TOE boundary that support the TOE functionality and security policy.
Protection Profile (PP)	An implementation-independent set of security requirements for a category of products.
Protection Profile Configuration	A comprehensive set of security requirements for a product type that consists of at least one Base-PP and at least one PP-Module.
Protection Profile Module (PP-Module)	An implementation-independent statement of security needs for a TOE type complementary to one or more Base Protection Profiles.
Security Assurance Requirement (SAR)	A requirement to assure the security of theTOE.
Security Functional Requirement (SFR)	A requirement for security enforcement by the TOE.
Security Target (ST)	A set of implementation-dependent security requirements for a specific product.
TOE Security Functionality (TSF)	The security functionality of the product under evaluation.
TOE Summary Specification (TSS)	A description of how a TOE satisfies the SFRs in a ST.
Target of Evaluation (TOE)	The product under evaluation.

1.2.2 Technical Terms

Administrator	Administrators perform management activities on the platform. These management functions do not include administration of tenant software running on the platform. Generally only server platforms have administrators that act through a Service Processor. Administrators may be no more than software entities.
Auditor	Auditors are responsible for managing the audit capabilities of theTOE. An Auditor may also be an Administrator. It is not a requirement that the TOE be capable of supporting an Auditor role that is separate from that of an Administrator.
Domain	A Domain or Information Domain is a policy construct that groups together execution environments and networks by sensitivity of information and access control policy. For example, classification levels represent information domains. Within classification levels, there might be other domains representing communities of interest or coalitions.
Information Domain	See Domain.
Introspection	A capability that allows a specially designated and privileged domain to have visibility into another domain for purposes of anomaly detection or monitoring.

Platform	The hardware and firmware on which tenant software executes. For thisPP, the TOE is the platform, hence there is no "TOE Platform."
System Security Policy (SSP)	The overall policy enforced by the TOE defining constraints on the behavior of the platform and on tenant software.
User	Users are entities that operate within the environment created by theTOE. The two kinds of users are Administrators and Tenants. Users need not be human. As far as the TOE is concerned, all users are just software.

1.3 Compliant Targets of Evaluation

A platform is a collection of hardware devices and firmware that provide the functional capabilities and services needed by tenant software. Such devices typically include embedded controllers, trusted platform modules, management controllers, host processors, network interface controllers, graphics processing units, flash memory, storage controllers, storage devices, boot firmware, runtime firmware, and a power supply.

A general-purpose computing platform is a hardware device that it capable of hosting more than one different operating system, virtualization system, or bare-metal application. Typical platform implementations include servers, PC clients, laptops, and tablets.

The PP does not apply to mobile devices (as defined by the Protection Profile for Mobile Device Fundamentals), network devices (as defined by the Collaborative Protection Profile for Network Devices), or to IoT devices.

1.3.1 TOE Boundary

The TOE is composed of hardware and firmware capable of running general-purpose software. This typically includes motherboards, embedded controllers, trusted platform modules, management controllers, host processors, network interface controllers, graphics processing units, flash memory, storage controllers, storage devices, boot firmware, runtime firmware, and a power supply.

The TOE does not include tenant software and customer workloads such as operating systems, virtualization systems, and applications. Firmware that comes pre-installed on the platform is considered to be part of the platform--whether its purpose is to support the functioning of the platform itself or to support tenant workloads. For example, a server's Service Processor and its firmware are considered part of the server platform.

1.3.2 Scope of Certification

Successful evaluation of a General-Purpose Computing Platform against this profile does not constitute or imply successful evaluation of any tenant software no matter how tightly integrated with the platform.

1.3.3 Product and Platform Equivalence

The tests in this Protection Profile must be run for all Platforms with which the Vendor would like to claim compliance—subject to this Profile’s equivalency guidelines (see Annex E).

1.4 Use Cases

This PP defines use cases for server platforms and client platforms:

Client Use Case:

Client platforms are computers that run an operating system or virtualization system on behalf of a user that is generally physically present at the machine. Examples include tablet computers, laptops, and desktop computers or workstations.

Server Use Case:

Servers platforms are computers that run an operating system or virtualization system in a data center or server environment on behalf of one or more users, or an enterprise. Generally, there is no user physically present at the machine. Server platforms often include integrated management subsystems.

2 Conformance Claims

Conformance Statement

An ST must claim exact conformance to thisPP, as defined in theCC and CEM addenda for Exact Conformance, Selection-Based SFRs, and Optional SFRs (dated May 2017).

CC Conformance Claims

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

PP Claim

This PP does not claim conformance to any Protection Profile.

Package Claim

This PP is TLS Package Conformant.

3 Security Problem Description

3.1 Threats

T.PLATFORM_COMPROMISE
The TOE must be capable of protecting itself from threats that originate from tenant software and operational networks connected to the TOE. The hosting of untrusted or malicious tenant software by theGPCP cannot be permitted to compromise the security and integrity of the platform.

3.2 Assumptions

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

A.TRUSTED_ADMIN
Administrators are trusted to follow and apply all administrator guidance.

A.COVERT_CHANNELS
If the has covert storage or timing channels, then for all tenant software executing on that TOE, it is assumed that relative to the IT assets to which they have access, the tenant software will have assurance sufficient to outweigh the risk that they will violate the security policy of the TOE by using those covert channels.

3.3 Organizational Security Policies

There are no organizational security policies defined for this PP.

4 Security Objectives

4.1 Security Objectives for the TOE

O.PLATFORM_INTEGRITY

Since the TOE is a platform, this objective probably shouldn't be here. So this is just a placeholder. The integrity of the TOE depends on the integrity of the software on which theTOE relies. Although the TOE does not have complete control over the integrity of the platform, the TOE should as much as possible try to ensure that no users or software hosted by the TOE is capable of undermining the integrity of the platform.

Addressed by: FDP_HBI_EXT.1, FDP_PPR_EXT.1, FDP_VMS_EXT.1, FDP_VNC_EXT.1, FPT_DVD_EXT.1, FPT_EEM_EXT.1, FPT_HAS_EXT.1, FPT_HAS_EXT.1, FPT_HCL_EXT.1, FPT_VDP_EXT.1, FPT_VIV_EXT.1

4.2 Security Objectives for the Operational Environment

OE.CONFIG

administrators will configure the Virtualization System correctly to create the intended security policy.

OE.PHYSICAL

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

OE.TRUSTED_ADMIN

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

OE.COVERT_CHANNELS

If the has covert storage or timing channels, then for all s executing on that , it is assumed that those s will have sufficient assurance relative to the IT assets to which they have access, to outweigh the risk that they will violate the security policy of the by using those covert channels.

OE.NON_MALICIOUS_USER

Users are trusted to be not willfully negligent or hostile and use theTOE in compliance with the applied enterprise security policy and guidance.

5 Security Requirements

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

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

5.1 TOE Security Functional Requirements

The Security Functional Requirements (SFRs) included in this section are derived from Part 2 of the Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 4, with additional extended functional components.

The defines operations on Security Functional Requirements: assignments, selections, assignments within selections and refinements.

5.1.1 Security Audit (FAU)

FAU_GEN.1 Audit Data Generation

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

a. Start-up and shutdown of audit functions
b. All administrative actions
c. [all auditable events defined in Table 1]
d. [auditable events defined in Table 2 for included SFRs]
e. [auditable events defined in Table 4 for included SFRs]
f. [auditable events defined in Table 5 for included SFRs]
g. [**selection**: *all auditable events defined in Table 3*, *no other auditable events*]

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

a. Date and time of the event
b. Type of event
c. Subject and object identity (if applicable)
d. The outcome (success or failure) of the event
e. [Additional information defined in Table 1]
f. [Additional information defined in Table 2 for included SFRs]
g. [Additional information defined in Table 4 for included SFRs]
h. [Additional information defined in Table 5 for included SFRs]
i. [**selection**: *Additional information defined in Table 3*, *no other information*]

Application Note: The author can include other auditable events directly in Table 1; they are not limited to the list presented. The author should update the table in FAU_GEN.1.2 with any additional information generated. “Subject identity” in FAU_GEN.1.2 could be a user id or an identifier specifying a , for example.

If ‘additional information defined in Table 3’ is selected, it is acceptable to include individual entries from Table 3 without including the entirety of Table 3. Appropriate entries from Tables 2, 4, and 5 should be included in the if the associated SFRs and selections are included.

The Table 1 entry for FDP_VNC_EXT.1 refers to configuration settings that attach s to virtualized network components. Changes to these configurations can be made during execution or when s are not running. Audit records must be generated for either case.

The intent of the audit requirement for FDP_PPR_EXT.1 is to log that the is connected to a physical device (when the device becomes part of the ’s hardware view), not to log every time that the device is accessed. Generally, this is only once at startup. However, some devices can be connected and disconnected during operation (e.g., virtual USB devices such as CD-ROMs). All such connection/disconnection events must be logged.

Evaluation Activity

TSS
The evaluator shall check the and ensure that it lists all of the auditable events and provides a format for audit records. Each audit record format type shall be covered, along with a brief description of each field. The evaluator shall check to make sure that every audit event type mandated by the PP is described in the.

Guidance
The evaluator shall also make a determination of the administrative actions that are relevant in the context of this PP. The evaluator shall examine the administrative guide and make a determination of which administrative commands, including subcommands, scripts, and configuration files, are related to the configuration (including enabling or disabling) of the mechanisms implemented in the that are necessary to enforce the requirements specified in the PP. The evaluator shall document the methodology or approach taken while determining which actions in the administrative guide are security-relevant with respect to this PP.

Tests
The evaluator shall test the 's ability to correctly generate audit records by having the generate audit records for the events listed and administrative actions. For administrative actions, the evaluator shall test that each action determined by the evaluator above to be security relevant in the context of this PP is auditable. When verifying the test results, the evaluator shall ensure the audit records generated during testing match the format specified in the administrative guide, and that the fields in each audit record have the proper entries.

Note that the testing here can be accomplished in conjunction with the testing of the security mechanisms directly.
See Table Table Table Table 1: Auditable Events for more information.

Requirement	***Auditable Events***	***Additional Audit Record Contents***
FAU_GEN.1	*None.*	*None.*
FCS_CKM.1	*None.*	*None.*
FDP_RIP_EXT.1	*None.*	*None.*
FIA_UAU.5	*None.*	*None.*
FMT_SMO_EXT.1	*Initiation of software update.*	*Version of update.*
FPT_HAS_EXT.1	*None.*	*None.*
FPT_ML_EXT.1	*No events specified*	
FTA_TAB.1	*None.*	*None.*

Table 1: Auditable Events

5.1.2 Cryptographic Support

FCS_CKM.1 Cryptographic Key Generation

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

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

].

Application Note: The author selects 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 shall 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 acts as a receiver in the RSA key establishment scheme, the does not need to implement RSA key generation.

Evaluation Activity

TSS
The evaluator shall ensure that the identifies the key sizes supported by the . If the specifies more than one scheme, the evaluator shall examine the to verify that it identifies the usage for each scheme.
Guidance
The evaluator shall verify that the AGD guidance instructs the administrator how to configure the to use the selected key generation scheme(s) and key size(s) for all uses defined in this PP.
Tests
Note: The following tests require the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on factory products.

Key Generation for FIPS PUB 186-4 RSA Schemes
The evaluator shall verify the implementation of RSA Key Generation by the using the Key Generation test. This test verifies the ability of the to correctly produce values for the key components including the public verification exponent e, the private prime factors p and q, the public modulus n and the calculation of the private signature exponent d.

Key Pair generation specifies 5 ways (or methods) to generate the primes p and q. These include:

- *Random Primes:*
  - *Provable primes*
  - *Probable primes*
- *Primes with Conditions:*
  - *Primes p1, p2, q1,q2, p and q shall all be provable primes*
  - *Primes p1, p2, q1, and q2 shall be provable primes and p and q shall be probable primes*
  - *Primes p1, p2, q1,q2, p and q shall all be probable primes*

To test the key generation method for the Random Provable primes method and for all the Primes with Conditions methods, the evaluator shall seed the key generation routine with sufficient data to deterministically generate the RSA key pair. This includes the random seed(s), the public exponent of the RSA key, and the desired key length. For each key length supported, the evaluator shall have the generate 25 key pairs. The evaluator shall verify the correctness of the 's implementation by comparing values generated by the with those generated from a known good implementation.

Key Generation for Elliptic Curve Cryptography (ECC)

FIPS 186-4 ECC Key Generation Test

For each supported NIST curve, i.e., P-256, P-384 and P-521, the evaluator shall require the implementation under test (IUT) to generate 10 private/public key pairs. The private key shall be generated using an approved random bit generator (RBG). To determine correctness, the evaluator shall submit the generated key pairs to the public key verification (PKV) function of a known good implementation.
FIPS 186-4 Public Key Verification (PKV) Test

For each supported NIST curve, i.e., P-256, P-384 and P-521, the evaluator shall generate 10 private/public key pairs using the key generation function of a known good implementation and modify five of the public key values so that they are incorrect, leaving five values unchanged (i.e., correct). The evaluator shall obtain in response a set of 10 PASS/FAIL values.

Key Generation for Finite-Field Cryptography (FFC)

The evaluator shall verify the implementation of the Parameters Generation and the Key Generation for FFC by the using the Parameter Generation and Key Generation test.

This test verifies the ability of the to correctly produce values for the field prime p, the cryptographic prime q (dividing p-1), the cryptographic group generator g, and the calculation of the private key x and public key y.

The Parameter generation specifies 2 ways (or methods) to generate the cryptographic prime q and the field prime p:

- *Primes q and p shall both be provable primes*
- *Primes q and field prime p shall both be probable primes*

and two ways to generate the cryptographic group generator g:

- *Generator g constructed through a verifiable process*
- *Generator g constructed through an unverifiable process.*

The Key generation specifies 2 ways to generate the private key x:

- *len(q) bit output of RBG where 1 □ x □ q-1*
- *len(q) + 64 bit output of RBG, followed by a mod q-1 operation where 1 □ x □ q-1*

The security strength of the RBG shall be at least that of the security offered by the FFC parameter set.

To test the cryptographic and field prime generation method for the provable primes method and/or the group generator g for a verifiable process, the evaluator shall seed the parameter generation routine with sufficient data to deterministically generate the parameter set.

For each key length supported, the evaluator shall have the generate 25 parameter sets and key pairs. The evaluator shall verify the correctness of the 's implementation by comparing values generated by the with those generated from a known good implementation. Verification shall also confirm

- *g != 0,1*
- *q divides p-1*
- *g^q mod p = 1*
- *g^x mod p = y*

for each FFC parameter set and key pair.

5.1.3 User Data Protection

FDP_RIP_EXT.1 Residual Information in Memory

FDP_RIP_EXT.1.1 The shall ensure that any previous information content of physical memory is cleared prior to allocation to a Guest .

Application Note: Physical memory must be zeroed before it is made accessible to afor general use by a Guest OS.

The purpose of this requirement is to ensure that a does not receive memory containing data previously used by another or the host.

"For general use" means for use by the Guest OS in its page tables for running applications or system software.

This does not apply to pages shared by design or policy between s or between the s and s, such as read-only OS pages or pages used for virtual device buffers.

Evaluation Activity

TSS
The evaluator shall ensure that the documents the process used for clearing physical memory prior to allocation to a Guest , providing details on when and how this is performed. Additionally, the evaluator shall ensure that the documents the conditions under which physical memory is not cleared prior to allocation to a Guest , and describes when and how the memory is cleared.

5.1.4 Identification and Authentication

FIA_UAU.5 Multiple Authentication Mechanisms

FIA_UAU.5.1 The shall provide the following authentication mechanisms: [**selection**:

- *[**selection**: local, directory-based] authentication based on username and password* ,
- *authentication based on username and a PIN that releases an asymmetric key stored in OE-protected storage*,
- *[**selection**: local, directory-based] authentication based on X.509 certificates* ,
- *[**selection**: local, directory-based] authentication based on an SSH public key credential*

] to support Administrator authentication.

Application Note: Selection of 'authentication based on username and password' requires that FIA_PMG_EXT.1 be included in the . This also requires that the include a management function for password management. If the author selects 'authentication based on an SSH public-key credential', the shall be validated against the Extended Package for Secure Shell.

PINs used to access OE-protected storage are set and managed by the OE-protected storage mechanism. Thus requirements on PIN management are outside the scope of the .

FIA_UAU.5.2 The shall authenticate any Administrator's claimed identity according to the [**assignment**: *rules describing how the multiple authentication mechanisms provide authentication*].

Evaluation Activity

Tests
If 'username and password authentication' is selected, the evaluator will configure the TOE with a known username and password and conduct the following tests:

- ***Test 1:** The evaluator will attempt to authenticate to theTOE using the known username and password. The evaluator will ensure that the authentication attempt is successful.*
- ***Test 2:** The evaluator will attempt to authenticate to theTOE using the known username but an incorrect password. The evaluator will ensure that the authentication attempt is unsuccessful.*

If 'username and PIN that releases an asymmetric key' is selected, the evaluator will examine the for guidance on supported protected storage and will then configure the or OE to establish a PIN which enables release of the asymmetric key from the protected storage (such as a TPM, a hardware token, or isolated execution environment) with which the TOE can interface. The evaluator will then conduct the following tests:

- ***Test 1:** The evaluator will attempt to authenticate to theTOE using the known user name and PIN. The evaluator will ensure that the authentication attempt is successful.*
- ***Test 2:** The evaluator will attempt to authenticate to theTOE using the known user name but an incorrect PIN. The evaluator will ensure that the authentication attempt is unsuccessful.*

If 'X.509 certificate authentication' is selected, the evaluator will generate an X.509v3 certificate for an Administrator user with the Client Authentication Enhanced Key Usage field set. The evaluator will provision the TOE for authentication with the X.509v3 certificate. The evaluator will ensure that the certificates are validated by the TOE as per FIA_X509_EXT.1.1 and then conduct the following tests:

- ***Test 1:** The evaluator will attempt to authenticate to theTOE using the X.509v3 certificate. The evaluator will ensure that the authentication attempt is successful.*
- ***Test 2:** The evaluator will generate a second certificate identical to the first except for the public key and any values derived from the public key. The evaluator will attempt to authenticate to the TOE with this certificate. The evaluator will ensure that the authentication attempt is unsuccessful.*

If 'SSH public-key credential authentication' is selected, the evaluator shall generate a public-private host key pair on the using RSA or ECDSA, and a second public-private key pair on a remote client. The evaluator shall provision the TOE with the client public key for authentication over SSH, and conduct the following tests:

- ***Test 1:** The evaluator will attempt to authenticate to theTOE using a message signed by the client private key that corresponds to provisioned client public key. The evaluator will ensure that the authentication attempt is successful.*
- ***Test 2:** The evaluator will generate a second client key pair and will attempt to authenticate to the TOE with the private key over SSH without first provisioning the TOE to support the new key pair. The evaluator will ensure that the authentication attempt is unsuccessful.*

5.1.5 Security Management

FMT_SMO_EXT.1 Separation of Management and Operational Networks

FMT_SMO_EXT.1.1 The shall support the configuration of separate management and operational networks through [**selection**: *physical means*, *logical means*, *trusted channel*].

FMT_SMO_EXT.1.2 **Application Note:** Management communications must be separate from user workloads. Administrative communications—including communications between physical hosts concerning load balancing, audit data, startup and shutdown—must be separate from guest operational networks.

"Physical means" refers to using separate physical networks for management and operational networks. For example, the machines in the management network are connected by separate cables plugged into separate and dedicated physical ports on each physical host.

"Logical means" refers to using separate network cables to connect physical hosts together using general-purpose networking ports. The management and operational networks are kept separate within the hosts using separate virtualized networking components.

If the author selects "trusted channel", then the protocols used for network separation must be selected in FTP_ITC_EXT.1.

Evaluation Activity

TSS
The evaluator shall examine the to verify that it describes how management and operational networks may be separated.
Guidance
The evaluator shall examine the operational guidance to verify that it details how to configure the TOE with separate Management and Operational Networks.
Tests
The evaluator shall configure the management network as documented. If separation is cryptographic or logical, then the evaluator shall capture packets on the management network. If Guest network traffic is detected, the requirement is not met.

5.1.6 Protection of the TSF

FPT_HAS_EXT.1 Hardware Assists

FPT_HAS_EXT.1.1 The shall use [**assignment**: *list of hardware-based virtualization assists*] to reduce or eliminate the need for binary translation.

FPT_HAS_EXT.1.2 The shall use [**assignment**: *list of hardware-based virtualization memory-handling assists*] to reduce or eliminate the need for shadow page tables.

Application Note: These hardware-assists help reduce the size and complexity of the , and thus, of the trusted computing base, by eliminating or reducing the need for paravirtualization or binary translation. Paravirtualization involves modifying guest software so that instructions that cannot be properly virtualized are never executed on the physical processor.

For the assignment in FPT_HAS_EXT.1, the author lists the hardware-based virtualization assists on all platforms included in the that are used by the to reduce or eliminate the need for software-based binary translation. Examples for the x86 platform are Intel VT-x and AMD-V. "None" is an acceptable assignment for platforms that do not require virtualization assists in order to eliminate the need for binary translation. This must be documented in the .

For the assignment in FPT_HAS_EXT.1.2, the author lists the set of hardware-based virtualization memory-handling extensions for all platforms listed in the that are used by the to reduce or eliminate the need for shadow page tables. Examples for the x86 platform are Intel EPT and AMD RVI. "None" is an acceptable assignment for platforms that do not require memory-handling assists in order to eliminate the need for shadow page tables. This must be documented in the .

Evaluation Activity

TSS
The evaluator shall examine the to ensure that it states, for each platform listed in the , the hardware assists and memory-handling extensions used by the on that platform. The evaluator shall ensure that these lists correspond to what is specified in the applicable FPT_HAS_EXT component.

5.1.7 TOE Access

FTA_TAB.1 TOE Access Banner

FTA_TAB.1.1 Before establishing an administrative user session, the shall display a security Administrator-specifiedad advisory notice and consent warning message regarding use of the .

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

Evaluation Activity

Tests
The evaluator shall configure the to display the advisory warning message "TEST TEST Warning Message TEST TEST". The evaluator shall then log out and confirm that the advisory message is displayed before logging can occur.

5.2 TOE Security Assurance Requirements

The Security Objectives for the in Section 4 were constructed to address threats identified in Section 3.1. The Security Functional Requirements (SFRs) in Section 5.1 are a formal instantiation of the Security Objectives. The PP 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 Security Assurance Requirements (SARs) from Part 3 of the Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 4 that are required in evaluations against this PP. Individual assurance activities to be performed are specified in both Section 5.1 as well as in this section.

After the has been approved for evaluation, the Information Technology Security Evaluation Facility (ITSEF) will obtain the , supporting environmental IT, and the administrative/user guides for the . The ITSEF is expected to perform actions mandated by the for the ASE and ALC SARs. The ITSEF also performs the assurance activities contained within Section 5, which are intended to be an interpretation of the other assurance requirements as they apply to the specific technology instantiated in the . The assurance activities that are captured in Section 5 also provide clarification as to what the developer needs to provide to demonstrate the is compliant with the PP.

5.2.1 Class ASE: Security Target Evaluation

As per ASE activities defined in [] plus the assurance activities defined for any SFRs claimed by the .

5.2.2 Class ADV: Development

The information about the is contained in the guidance documentation available to the end user as well as the Summary Specification () portion of the . The developer must concur with the description of the product that is contained in the as it relates to the functional requirements. The Assurance Activities contained in Section 5.2 should provide the authors with sufficient information to determine the appropriate content for the section.

ADV_FSP.1 Basic functional specification

Developer action elements:

ADV_FSP.1.1 The developer shall provide a functional specification.

ADV_FSP.1.2 The developer shall provide a tracing from the functional specification to the SFRs.

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

Content and presentation elements:

ADV_FSP.1.3 The functional specification shall describe the purpose and method of use for each SFR-enforcing and SFR-supporting TSFI.

ADV_FSP.1.4 The functional specification shall identify all parameters associated with each SFR-enforcing and SFR-supporting TSFI.

ADV_FSP.1.5 The functional specification shall provide rationale for the implicit categorization of interfaces as SFR-non-interfering.

ADV_FSP.1.6 The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

Evaluator action elements:

ADV_FSP.1.7 The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_FSP.1.8 The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

Application Note: There are no specific assurance activities associated with these SARs. The functional specification documentation is provided to support the evaluation activities described in Section 5.2, and other activities described for AGD, ATE, and AVA SARs. The requirements on the content of the functional specification information is implicitly assessed by virtue of the other assurance activities being performed; if the evaluator is unable to perform an activity because the there is insufficient interface information, then an adequate functional specification has not been provided.

5.2.3 Class AGD: Guidance Documents

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

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

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

Guidance pertaining to particular security functionality is also provided; specific requirements on such guidance are contained in the assurance activities specified with individual SFRs where applicable.

AGD_OPE.1 Operational User Guidance

Developer action elements:

AGD_OPE.1.1 The developer shall provide operational user guidance.

Developer Note: Rather than repeat information here, the developer should review the assurance activities for this component to ascertain the specifics of the guidance that the evaluators will be checking for. This will provide the necessary information for the preparation of acceptable guidance.

Content and presentation elements:

AGD_OPE.1.2 The operational user guidance shall describe what the authorized user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.

AGD_OPE.1.3 The operational user guidance shall describe, for the authorized user, how to use the available interfaces provided by the in a secure manner.

AGD_OPE.1.4 The operational user guidance shall describe, for the authorized user, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.

AGD_OPE.1.5 The operational user guidance shall, for the authorized user, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the .

AGD_OPE.1.6 The operational user guidance shall identify all possible modes of operation of the (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AGD_OPE.1.7 The operational user guidance shall, for the authorized user, describe the security measures to be followed in order to fulfill the security objectives for the operational environment as described in the .

AGD_OPE.1.8 The operational user guidance shall be clear and reasonable.

Evaluator action elements:

AGD_OPE.1.9 The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

Evaluation Activity

Guidance
Some of the contents of the operational guidance will be verified by the assurance activities in Section 5.2 and evaluation of the according to the . The following additional information is also required.

The operational guidance shall contain instructions for configuring the password characteristics, number of allowed authentication attempt failures, the lockout period times for inactivity, and the notice and consent warning that is to be provided when authenticating.

The operational guidance shall contain step-by-step instructions suitable for use by an end-user of the Virtualization System to configure a new, out-of-the-box system into the configuration evaluated under this Protection Profile.

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

- *Instructions for querying the current version of thesoftware.*
- *For hashes, a description of where the hash for a given update can be obtained. For digital signatures, instructions for obtaining the certificate that will be used by the FCS_COP.1(2) mechanism to ensure that a signed update has been received from the certificate owner. This may be supplied with the product initially, or may be obtained by some other means.*
- *Instructions for obtaining the update itself. This should include instructions for making the update accessible to the (e.g., placement in a specific directory).*
- *Instructions for initiating the update process, as well as discerning whether the process was successful or unsuccessful. This includes generation of the hash/digital signature.*

AGD_PRE.1 Preparative procedures

Developer action elements:

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

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

Content and presentation elements:

AGD_PRE.1.2 The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered in accordance with the developer's delivery procedures.

AGD_PRE.1.3 The preparative procedures shall describe all the steps necessary for secure installation of the and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the .

Evaluator action elements:

AGD_PRE.1.4 The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AGD_PRE.1.5 The evaluator shall apply the preparative procedures to confirm that the can be prepared securely for operation.

Evaluation Activity

Guidance
As indicated in the introduction above, there are significant expectations with respect to the documentation—especially when configuring the operational environment to support functional requirements. The evaluator shall check to ensure that the guidance provided for the adequately addresses all platforms (that is, combination of hardware and operating system) claimed for the in the .

The operational guidance shall contain step-by-step instructions suitable for use by an

end-user of the Virtualization System to configure a new, out-of-the-box system into the configuration evaluated under this Protection Profile.

5.2.4 Class ALC: Life-Cycle Support

At the assurance level specified for s conformant to this PP, life-cycle support is limited to an examination of the vendor's development and configuration management process in order to provide a baseline level of assurance that the itself is developed in a secure manner and that the developer has a well-defined process in place to deliver updates to mitigate known security flaws. This is a result of the critical role that a developer's practices play in contributing to the overall trustworthiness of a product.

ALC_CMC.1 Labeling of the TOE

Developer action elements:

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

Content and presentation elements:

ALC_CMC.1.2 The shall be labeled with its unique reference.

Evaluator action elements:

ALC_CMC.1.3 The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

Evaluation Activity

TSS
The evaluator shall check the ST to ensure that it contains an identifier (such as a product name/version number) that specifically identifies the version that meets the requirements of the ST.
Guidance
The evaluator shall check the AGD guidance and TOE samples received for testing to ensure that the version number is consistent with that in the ST.
Tests
If the vendor maintains a web site advertising the TOE, the evaluator shall examine the information on the web site to ensure that the information in the ST is sufficient to distinguish the product.

ALC_CMS.1 TOE CM coverage

Developer action elements:

ALC_CMS.1.1 The developer shall provide a configuration list for the .

Content and presentation elements:

ALC_CMS.1.2 The configuration list shall include the following: the itself; and the evaluation evidence required by the SARs.

ALC_CMS.1.3 The configuration list shall uniquely identify the configuration items.

Evaluator action elements:

ALC_CMS.1.4 The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

Evaluation Activity

Guidance
The evaluator shall ensure that the developer has identified (in public-facing development guidance for their platform) one or more development environments appropriate for use in developing applications for the developer's platform. For each of these development environments, the developer shall provide information on how to configure the environment to ensure that buffer overflow protection mechanisms in the environment(s) are invoked (e.g., compiler and linker flags). The evaluator shall ensure that this documentation also includes an indication of whether such protections are on by default, or have to be specifically enabled. The evaluator shall ensure that the TSF is uniquely identified (with respect to other products from the TSF vendor), and that documentation provided by the developer in association with the requirements in the ST

is associated with the TSF using this unique identification.

ALC_TSU_EXT.1 Timely Security Updates

Developer action elements:

ALC_TSU_EXT.1.1 The developer shall provide a description in the of how timely security updates are made to the .

Content and presentation elements:

ALC_TSU_EXT.1.2 The description shall include the process for creating and deploying security updates for the software/firmware.

ALC_TSU_EXT.1.3 The description shall express the time window as the length of time, in days, between public disclosure of a vulnerability and the public availability of security updates to the .

Application Note: The total length of time may be presented as a summation of the periods of time that each party (e.g., developer, hardware vendor) on the critical path consumes. The time period until public availability per deployment mechanism may differ; each is described.

ALC_TSU_EXT.1.4 The description shall include the mechanisms publicly available for reporting security issues pertaining to the .

Application Note: The reporting mechanism could include web sites, email addresses, and a means to protect the sensitive nature of the report (e.g., public keys that could be used to encrypt the details of a proof-of-concept exploit).

Evaluator action elements:

ALC_TSU_EXT.1.5 The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.5 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 ATE_IND family, while the latter is through the AVA_VAN family. At the assurance level specified in this PP, testing is based on advertised functionality and interfaces with dependency on the availability of design information. One of the primary outputs of the evaluation process is the test report as specified in the following requirements.

ATE_IND.1 Independent Testing - Sample

Developer action elements:

ATE_IND.1.1 The developer shall provide the for testing.

Content and presentation elements:

ATE_IND.1.2 The shall be suitable for testing.

Evaluator action elements:

ATE_IND.1.3 The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.1.4 The evaluator shall test a subset of the to confirm that the operates as specified.

Evaluation Activity

Tests
The evaluator shall prepare a test plan and report documenting the testing aspects of the system. While it is not necessary to have one test case per test listed in an Assurance Activity, the evaluators must document in the test plan that each applicable testing requirement in the is covered.

The Test Plan identifies the platforms to be tested, and for those platforms not included in the test plan but included in the , the test plan provides a justification for not testing the platforms. This justification must address the differences between the tested platforms and the untested platforms, and make an argument that the differences do not affect the testing to be performed. It is not sufficient to merely assert that the differences have no affect; rationale must be provided. If all platforms claimed in the are tested, then no rationale is necessary.

The test plan describes the composition of each platform to be tested, and any setup that is necessary beyond what is contained in the AGD documentation. It should be noted that the evaluators are expected to follow the AGD documentation for installation and setup of each platform either as part of a test or as a standard pre-test condition. This may include special test drivers or tools. For each driver or tool, an argument (not just an assertion) is provided that the driver or tool will not adversely affect the performance of the functionality by the and its platform. This also includes the configuration of cryptographic engines to be used. The cryptographic algorithms implemented by these engines are those specified by this PP and used by the cryptographic protocols being evaluated (IPsec, TLS/HTTPS, SSH).

The test plan identifies high-level test objectives as well as the test procedures to be followed to achieve those objectives. These procedures include expected results. The test report (which could just be an annotated version of the test plan) details the activities that took place when the test procedures were executed, and includes the actual results of the tests. This shall be a cumulative account, so if there was a test run that resulted in a failure; a fix installed; and then a successful re-run of the test, the report would show a "fail" and "pass" result (and the supporting details), and not just the "pass" result.

5.2.6 Class AVA: Vulnerability Assessment

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

AVA_VAN.1 Vulnerability survey

Developer action elements:

AVA_VAN.1.1 The developer shall provide the for testing.

Content and presentation elements:

AVA_VAN.1.2 The shall be suitable for testing.

Evaluator action elements:

AVA_VAN.1.3 The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_VAN.1.4 The evaluator shall perform a search of public domain sources to identify potential vulnerabilities in the .

AVA_VAN.1.5 The evaluator shall conduct penetration testing, based on the identified potential vulnerabilities, to determine that the is resistant to attacks performed by an attacker possessing Basic attack potential.

Evaluation Activity

As with ATE_IND the evaluator shall generate a report to document their findings with respect to this requirement. This report could physically be part of the overall test report mentioned in ATE_IND, or a separate document. The evaluator performs a search of public information to determine the vulnerabilities that have been found in virtualization in general, as well as those that pertain to the particular . The evaluator documents the sources consulted and the vulnerabilities found in the report. For each vulnerability found, the evaluator either provides a rationale with respect to its non-applicability or the evaluator formulates a test (using the guidelines provided in ATE_IND) to confirm the vulnerability, if suitable. Suitability is determined by assessing the attack vector needed to take advantage of the vulnerability. For example, if the vulnerability can be detected by pressing a key combination on boot-up, a test would be suitable at the assurance level of this PP. If exploiting the vulnerability requires expert skills and an electron microscope, for instance, then a test would not be suitable and an appropriate justification would be formulated.

Appendix A - Optional Requirements

As indicated in the introduction to this PP, the baseline requirements (those that must be performed by theTOE) are contained in the body of this PP. This Appendix contains three other types of optional requirements that may be included in the ST, but are not required in order to conform to this PP. However, applied modules, packages and/or use cases may refine specific requirements as mandatory.

The first type (A.1 Strictly Optional Requirements) are strictly optional requirements that are independent of theTOE implementing any function. If the TOE fulfills any of these requirements or supports a certain functionality, the vendor is encouraged to included the SFRs in the ST, but are not required in order to conform to thisPP.

The second type (A.2 Objective Requirements) are objective requirements that describe security functionality not yet widely available in commercial technology. The requirements are not currently mandated in the body of this PP, but will be included in the baseline requirements in future versions of this PP. Adoption by vendors is encouraged and expected as soon as possible.

The third type (A.3 Implementation-Dependent Requirements) are dependent on the TOE implementing a particular function. If the TOE fulfills any of these requirements, the vendor must either add the relatedSFR or disable the functionality for the evaluated configuration.

A.1 Strictly Optional Requirements

A.2 Objective Requirements

A.2.1 Protection of the TSF

FPT_ML_EXT.1 Measured Launch of Platform and VMM

FPT_ML_EXT.1.1 The shall support a measured launch of the Virtualization System. Measured components of the Virtualization system shall include the static executable image of the Hypervisor and: [**selection**:

- *Static executable images of the Management Subsystem*,
- *[**assignment**: list of (static images of) Service s]*,
- *[**assignment**: list of configuration files]*,
- *no other components*

]

FPT_ML_EXT.1.2 The shall make these measurements available to the Management Subsystem.

Application Note: A measured launch of the platform and Virtualization System, demonstrates that the proper software was loaded. A measured launch process employs verifiable integrity measurement mechanisms. For example, a TOE may hash components such as: the hypervisor, service s and/or the Management Subsystem. A measured launch process only allows components to be executed after the measurement has been recorded. An example process may add each component's hash before it is executed so that the final hash reflects the evidence of a component's state prior to execution. The measurement may be verified as the system boots, but this is not required.

The Platform is outside of the . However, this requirement specifies that theTOE must be capable of receiving Platform measurements if the Platform provides them. This requirement is requiring support for Platform measurements if provided; it is not placing a requirement on the Platform to take such measurements.

If available, hardware should be used to store measurements in such a manner that they cannot be modified in any manner except to be extended. These measurements should be produced in a repeatable manner so that a third party can verify the measurements if given the inputs. Hardware devices, like Trusted Platform Modules (TPM), TrustZone, and MMU are some examples that may serve as foundations for storing and reporting measurements.

Platforms with a root of trust for measurement (RTM) should initiate the measured launch process. This may include core BIOS or the chipset. The chipset is the preferred RTM, but core BIOS or other firmware is acceptable. In system without a traditional RTM, the first component that boots would be considered the RTM, this is not preferred.

Evaluation Activity

TSS

The evaluator shall verify that the or Operational Guidance describes how integrity measurements are performed and made available to the Management Subsystem. The evaluator shall examine the operational guidance to verify that it documents how to access the measurements in the Management Subsystem.
Tests
The evaluator shall perform the following tests:

- ***Test 1:*** *The evaluator shall start the TOE, login as an Administrator, and verify that the measurements for the specified components are viewable in the Management Subsystem.*

A.3 Implementation-Dependent Requirements

This PP does not define any implementation-dependent requirements.

Appendix B - Selection-Based Requirements

As indicated in the introduction to this PP, the baseline requirements (those that must be performed by theTOE 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 must be included.

This PP does not define any selection-based requirements.

Appendix C - Entropy Documentation and Assessment

C.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.

C.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 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.

C.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.

C.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.

Appendix D - Equivalency Guidlines

D.1 Introduction

The purpose of equivalence in PP-based evaluations is to find a balance between evaluation rigor and commercial practicability--to ensure that evaluations meet customer expectations while recognizing that there is little to be gained from requiring that every variation in a product or platform be fully tested. If a product is found to be compliant with a PP on one platform, then all equivalent products on equivalent platforms are also considered to be compliant with the PP.

A Vendor can make a claim of equivalence if the Vendor believes that a particular instance of their Product implements PP-specified security functionality in a way equivalent to the implementation of the same functionality on another instance of their Product on which the functionality was tested. The Product instances can differ in version number or feature level (model), or the instances may run on different platforms. Equivalency can be used to reduce the testing required across claimed evaluated configurations. It can also be used during Assurance Maintenance to reduce testing needed to add more evaluated configurations to a certification.

These equivalency guidelines do not replace Assurance Maintenance requirements or NIAP Policy #5 requirements for CAVP certificates. Nor may equivalency be used to leverage evaluations with expired certifications.

This document provides guidance for determining whether Products and Platforms are equivalent for purposes of evaluation against the Protection Profile for Virtualization (VPP) when instantiated with either the Client or Server Extended Package.

Equivalence has two aspects:

1. ***Product Equivalence:*** Products may be considered equivalent if there are no differences between Product Models and Product Versions with respect to PP-specified security functionality.
2. ***Platform Equivalence:*** Platforms may be considered equivalent if there are no significant differences in the services they provide to the Product--or in the way the platforms provide those services--with respect to PP-specified security functionality.

The equivalency determination is made in accordance with these guidelines by the Validator and Scheme using information provided by the Evaluator/Vendor.

D.2 Approach to Equivalency Analysis

There are two scenarios for performing equivalency analysis. One is when a product has been certified and the vendor wants to show that a later product should be considered certified due to equivalence with the earlier product. The other is when multiple product variants are going though evaluation together and the vendor would like to reduce the amount of testing that must be done. The basic rules for determining equivalence are the same in both cases. But there is one additional consideration that applies to equivalence with previously certified products. That is, the product with which equivalence is being claimed must have a valid certification in accordance with scheme rules and the Assurance Maintenance process must be followed. If a product's certification has expired, then equivalence cannot be claimed with that product.

When performing equivalency analysis, the Evaluator/Vendor should first use the factors and guidelines for Product Model equivalence to determine the set of Product Models to be evaluated. In general, Product Models that do not differ in PP-specified security functionality are considered equivalent for purposes of evaluation against the VPP.

If multiple revision levels of Product Models are to be evaluated--or to determine whether a revision of an evaluated product needs re-evaluation--the Evaluator/Vendor and Validator should use the factors and guidelines for Product Version equivalence to determine whether Product Versions are equivalent.

Having determined the set of Product Models and Versions to be evaluated, the next step is to determine the set of Platforms that the Products must be tested on.

Each non-equivalent Product for which compliance is claimed must be fully tested on each non-equivalent platform for which compliance is claimed. For non-equivalent Products on equivalent platforms, only the differences that affect PP-specified security functionality must be tested for each product.

If the set of equivalent Products includes only bare-metal installations, then the equivalency analysis is complete. But if any members of the set include hosted installations or installations that integrate with an existing host operating system or control domain, then software platform equivalence must be taken into consideration. The Evaluator/Vendor and Validator should use the factors and guidance for software platform equivalence to determine whether different models or versions of host or control domain operating systems require separate testing. ***"Differences in PP-Specified Security Functionality" Defined***
If PP-specified security functionality is implemented by the TOE, then differences in the actual implementation between versions or product models break equivalence for that feature. Likewise, if the TOE implements the functionality in one version or model and the functionality is implemented by the platform in another version or model, then equivalence is broken. If the functionality is implemented by the platform in multiple models or versions on equivalent platforms, then the functionality is considered different if the product invokes the platform differently to perform the function.

D.3 Specific Guidance for Determining Product Model Equivalence

Product Model equivalence attempts to determine whether different feature levels of the same product across a product line are equivalent for purposes of PP testing. For example, if a product has a “basic” edition and an “enterprise” edition, is it necessary to test both models? Or does testing one model provide sufficient assurance that both models are compliant?

Table 1, below, lists the factors for determining Product Model equivalence.

Factor	Same/Different	Guidance
Target Platform	Different	Product Models that virtualize different instruction sets (e.g. x86, ARM, POWER, SPARC, MIPS) are not equivalent.
Installation Types	Different	If a Product can be installed either on bare metal or onto an operating system (either Type 1 or Type 2), and the vendor wants to claim that both installation types constitute a single Model, then see the guidance for “PP-Specified Functionality,” below.
Software Platform	Different	Product Models that run on substantially different software environments, such as different host operating systems, are not equivalent. Models that install on different versions of the same software environment may be equivalent depending on the below factors.
PP-Specified Functionality	Same	If the differences between Models affect only non-PP-specified functionality, then the Models are equivalent.
	Different	If PP-specified security functionality is affected by the differences between Models, then the Models are not equivalent and must be tested separately. It is necessary to test only the functionality affected by the software differences. If only differences are tested, then the differences must be enumerated, and for each difference the Vendor must provide an explanation of why each difference does or does not affect PP-specified functionality. If the Product Models are fully tested separately, then there is no need to document the differences.

Table 1. Factors for Determining Product Model Equivalance

D.4 Specific Guidance for Determining Product Version Equivalence

In cases of version equivalence, differences are expressed in terms of changes implemented in revisions of an evaluated Product. In general, versions are equivalent if the changes have no effect on any security-relevant claims about the TOE or assurance evidence. Non-security-relevant changes to TOE functionality or the addition of non-security-relevant functionality does not affect equivalence.

Factor	Same/Different	Guidance
Product Models	Different	Versions of different Product Models are not equivalent unless the Models are equivalent as defined in Section 3.
PP-Specified Functionality	Same	If the differences affect only non-PP-specified functionality, then the Versions are equivalent.
	Different	If PP-specified security functionality is affected by the differences, then the Versions are considered to be not equivalent and must be tested separately. It is necessary only to test the functionality affected by the changes. If only the differences are tested, then for each difference the Vendor must provide an explanation of why the difference does or does not affect PP-specified functionality. If the Product Versions are fully tested separately, then there is no need to document the differences.

Table 2. Factors for Determining Product Version Equivalence

D.5 Specific Guidance for Determining Platform Equivalence

Platform equivalence is used to determine the platforms that a product must be tested on. These guidelines are divided into sections for determining hardware equivalence and software (host OS/control domain) equivalence. If the Product is installed onto bare metal, then only hardware equivalence is relevant. If the Product is installed onto an OS—or is integrated into an OS—then both hardware and software equivalence are required. Likewise, if the Product can be installed either on bare metal or on an operating system, both hardware and software equivalence are relevant.

D.5.1 Hardware Platform Equivalence

If a Virtualization Solution runs directly on hardware without an operating system, then platform equivalence is based primarily on processor architecture and instruction sets.

Platforms with different processor architectures and instruction sets are not equivalent. This is probably not an issue because there is likely to be a different product model for different hardware environments.

Equivalency analysis becomes important when comparing platforms with the same processor architecture. Processors with the same architecture that have instruction sets that are subsets or supersets of each other are not disqualified from being equivalent for purposes of a VPP evaluation. If the VS takes the same code paths when executing PP-specified security functionality on different processors of the same family, then the processors can be considered equivalent with respect to that application.

For example, if a VS follows one code path on platforms that support the AES-NI instruction and another on platforms that do not, then those two platforms are not equivalent with respect to that VS functionality. But if the VS follows the same code path whether or not the platform supports AES-NI, then the platforms are equivalent with respect to that functionality.

The platforms are equivalent with respect to the VS if the platforms are equivalent with respect to all PP-specified security functionality.

Factor	Same/Different/None	Guidance
Platform Architectures	Different	Haardware platforms that implement different processor architectures and instruction sets are not equivalent.
PP-Specified Functionality	Same	For platforms with the same processor architecture, the platforms are equivalent with respect to the application if execution of all PP-specified security functionality follows the same code path on both platforms.

Table 3. Factors for Determining Hardware Platform Equivalence

D.5.2 Software Platform Equivalence

If the Product installs onto or integrates with an operating system that is not installed with the product--and thus is not part of the TOE--then the Product must be tested on all non-equivalent Software Platforms.

The guidance for Product Model (Section 3) specifies that Products intended for use on substantially different operating systems (e.g. Windows vs. Linux vs. SunOS) are different Models. Therefore, platforms running substantially different operating systems are de facto not equivalent. Likewise, operating systems with different major version numbers are not equivalent for purposes of this PP.

As a result, Software Platform equivalence is largely concerned with revisions and variations of operating systems that are substantially the same (e.g. different versions and revision levels of Windows or Linux).

Factor	Same/Different/None	Guidance
Platform Type/Vendor	Different	Operating systems that are substantially different or come from different vendors are not equivalent.
Platform Versions	Different	Operating systems are not equivalent if they have different major version numbers.
PP-Specified Functionality	Same	If the differences between software platform models or versions affect only non-PP-specified functionality, then the software platforms are equivalent.
	Different	If PP-specified security functionality is affected by the differences between software platform versions or models, then the software platforms are not considered equivalent and must be tested separately. It is necessary only to test the functionality affected by the changes. If only the differences are tested, then for each difference the Vendor must provide an explanation of why the difference does or does not affect PP-specified functionality. If the Products are fully tested on each platform, then there is no need to document the differences.

Table 4. Factors for Software Platform Equivalence

D.6 Level of Specificity for Tested Configurations and Claimed Equivalent Configurations

In order to make equivalency determinations, the vendor and evaluator must agree on the equivalency claims. They must then provide the scheme with sufficient information about the TOE instances and platforms that were evaluated, and the TOE instances and platforms that are claimed to be equivalent.

The ST must describe all configurations evaluated down to processor manufacturer, model number, and microarchitecture version.

The information regarding claimed equivalent configurations depends on the platform that the VS was developed for and runs on.

Bare-Metal VS

For VSes that run without an operating system on bare-metal or virtual bare-metal, the claimed configuration must describe the platform down to the specific processor manufacturer, model number, and microarchitecture version. The Vendor must describe the differences in the TOE with respect to PP-specified security functionality and how the TOE operates differently to leverage platform differences (e.g., instruction set extensions) in the tested configuration versus the claimed equivalent configuration.

VS with OS Support

For VSes that run on an OS host or with the assistance of an OS, then the claimed configuration must describe the OS down to its specific model and version number. The Vendor must describe the differences in the TOE with respect to PP-specified security functionality and how the TOE functions differently to leverage platform differences in the tested configuration versus the claimed equivalent configuration.

Appendix E - References

Identifier	Title
[CC]	Common Criteria for Information Technology Security Evaluation - • Part 1: Introduction and General Model, CCMB-2017-04-001, Version 3.1, Revision 5, April 2017. • Part 2: Security Functional Components, CCMB-2017-04-002, Version 3.1, Revision 5, April 2017. • Part 3: Security Assurance Components, CCMB-2017-04-003, Version 3.1, Revision 5, April 2017.
[CEM]	Common Evaluation Methodology for Information Technology Security - Evaluation Methodology, CCMB-2017-04-004, Version 3.1, Revision 5, April 2017.

Appendix F - Acronyms

Acronym	Meaning
AES	Advanced Encryption Standard
CC	Common Criteria
CEM	Common Evaluation Methodology
CPU	Central Processing Unit
DEP	Data Execution Prevention
DKM	Derived Keying Material
DSS	Digital Signature Standard
ECC	Elliptic Curve Cryptography
FFC	Finite-Field Cryptography
FIPS	Federal Information Processing Standard
GPCP	General-Purpose Computing Platform
IEC	International Electrotechnical Commission
IP	Internet Protocol
ISO	International Organization for Standardization
IT	Information Technology
ITSEF	Information Technology Security Evaluation Facility
KDF	Key Derivation Function
MAC	Message Authentication Code
NIST	National Institute of Standards and Technology
NVLAP	National Voluntary Laboratory Accreditation Program
OS	Operating System
PKV	Public Key Verification
PP	Protection Profile
PP-Module	Protection Profile Module
RSA	Rivest, Shamir, Adleman
SAR	Security Assurance Requirement
SFR	Security Functional Requirement
SP	Special Publication
SPD	Security Policy Database
SSP	System Security Policy
ST	Security Target
SWID	Software Identification
TOE	Target of Evaluation
TPM	Trusted Platform Module
TSF	TOE Security Functionality

TSS	TOE Summary Specification