Requirements from the

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

Introduction

Purpose. This document presents the functional and assurance requirements found in the *Protection Profile for General-Purpose Computing Platforms*. Common Criteria evaluation, facilitated in the U.S. by the National Information Assurance Partnership (NIAP), is required for IA and IA-enabled products in National Security Systems according to CNSS Policy #11.

Using this document. This representation of the Protection Profile includes:

- *Security Functional Requirements* for use in evaluation. These are featured without the formal Assurance Activities specified in the Protection Profile, to assist the reader who is interested only in the requirements.

  It also includes, in tables shown later, particular types of security functional requirements that are not strictly required in all cases. These are:

  - *Selection-based Security Functional Requirements* which become required when certain selections are made inside the regular Security Functionality Requirements (as indicated by the **[selection:]** construct).
  - *Objective Security Functional Requirements* which are highly desired but not yet widely-available in commercial technology.
  - *Optional Security Functional Requirements* which are available for evaluation and which some customers may insist upon.

- *Security Assurance Requirements* which relate to developer support for the product under evaluation, development processes, and other non-functionality security relevant requirements.

Security Functional Requirements

Audit Data Generation

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*]

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.

Cryptographic Key Generation

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.

Residual Information in Memory

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 a for 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.

Multiple Authentication Mechanisms

FIL-AUA 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 .

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

Separation of Management and Operational Networks

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

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.

Hardware Assists

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

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 .

TOE Access Banner

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.

Security Assurance Requirements

The developer shall provide a functional specification.

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

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

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

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

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

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

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.

The developer shall provide operational user guidance.

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.

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

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.

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 .

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.

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 .

The operational user guidance shall be clear and reasonable.

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

The developer shall provide the including its preparative procedures.

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

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 .

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

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

The developer shall provide the and a reference for the .

The shall be labeled with its unique reference.

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

The developer shall provide a configuration list for the .

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

The configuration list shall uniquely identify the configuration items.

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

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

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

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.

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).

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

The developer shall provide the for testing.

The shall be suitable for testing.

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

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

The developer shall provide the for testing.

The shall be suitable for testing.

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

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

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.

Selection-Based Security Functional Requirements

Objective Security Functional Requirements

Measured Launch of Platform and VMM

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*

]

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 the TOE 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.

Optional Security Functional Requirements