Tabular Presentation of the *Protection Profile for General-Purpose Computing Platforms*

Version: 0.1

2019-12-13

National Information Assurance Partnership

Revision History

Version	Date	Comment

Introduction

This document presents the Security Functional Requirements and Security Assurance Requirements from the *Protection Profile for General-Purpose Computing Platforms*. This tabular representation is provided for those audiences whose interest primarily lies in those portions of that document. The Protection Profile itself remains the only complete and authoritative representation, and includes discussion of assumptions, threats, and objectives.

Security Functional Requirements

ID	Requirement	Assurance Activity
	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.	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 . 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. The evaluator shall test the ’s ability to correctly generate audit records by

ID	Requirement	Assurance Activity
	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.	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 : Auditable Events for more information. : Auditable Events

	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.
		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. 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. 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

ID	Requirement	Assurance Activity
		• 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.
	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.	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.
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*].	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 the TOE 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 the TOE 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 the TOE 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 the TOE 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 the TOE 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

ID	Requirement	Assurance Activity
		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 the TOE 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.
	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.	The evaluator shall examine the to verify that it describes how management and operational networks may be separated. The evaluator shall examine the operational guidance to verify that it details how to configure the TOE with separate Management and Operational Networks. 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.
	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 .	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.
	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*] ***This is currently an objective requirement.***	
	The shall make these measurements available to the Management Subsystem. ***This is currently an objective requirement.*** **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	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. 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.

ID	Requirement	Assurance Activity
	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.	
	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.	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.

Security Assurance Requirements

ID	Requirement	Assurance Activity
	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.	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.

ID	Requirement	Assurance Activity
		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 the software. • 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.
	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.	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.
	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 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. 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. 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.
	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 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.
	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 evaluator shall prepare a test plan and report documenting the testing aspects of the system. While it is not

ID	Requirement	Assurance Activity
	the operates as specified.	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.
	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.	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.

Glossary