# Protection Profile for QQQQ

## This page is best viewed with JavaScript enabled!


Version: 1.0

2015-08-14

**National Information Assurance Partnership**

## Revision History

| Version | Date | Comment |
|---|---|---|
| Round 1 | 2015-04-23 | First draft of version 1.0 for comment |
| 1.0 | 2015-08-14 | Release - first version released |

## Contents

1Introduction1.1Overview1.2Terms1.2.1Common Criteria Terms1.2.2Technical Terms1.3Compliant Targets of Evaluation1.3.1TOE Boundary1.3.2TOE Platform1.4Use Cases2Conformance Claims3Security Problem Description3.1Threats3.2Assumptions4Security Objectives4.1Security Objectives for the TOE4.2Security Objectives for the Operational Environment4.3Security Objectives Rationale5Security Requirements5.1Security Functional Requirements5.1.1Something5.1.2QQQQ5.1.23QQQQ5.1.4Security Management (FMT)5.1.35Security Audit (FAU)5.1.46TOE Security Functional Requirements Rationale5.2Security Assurance Requirements5.2.1Class ASE: Security Target5.2.2Class ADV: Development5.2.3Class AGD: Guidance Documentation5.2.4Class ALC: Life-cycle Support5.2.5Class ATE: Tests5.2.6Class AVA: Vulnerability AssessmentAppendix A - Optional RequirementsA.1Strictly Optional Requirements A.1.1QQQQA.1.2Security Audit (FAU)A.2Objective Requirements A.2.1QQQQA.3Implementation-Dependent based Requirements A.3.1Widget ThingA.3.1.1QQQQAppendix B - Selection-Based based Requirements B.1QQQQAppendix C - Use Case TemplatesC.1Elephant-own deviceAppendix D - Inherently Satisfied RequirementsAppendix D - Entropy Documentation and AssessmentD.1Design DescriptionD.2Entropy JustificationD.3Operating ConditionsD.4Health TestingAppendix E - ReferencesAppendix F - Acronyms

# 1 Introduction

## 1.1 Overview

The scope of this Protection Profile (PP) is to describe the security functionality of QQQQ products in terms of[CC] and to define functional and assurance requirements for such products. An operating system is software that manages computer hardware and software resources, and provides common services for application programs. The hardware it manages may be physical, virtual or imaginary.
Something

This is going to show some tests:

- Terms with abbrs like ASLR, or API, should be found a linked automatically.
- And components can be refered to by their name: FQQ_QQQ.1
- And so can requirements: FQQ_QQQ.1.1 or by their unique identifier: FQQ_QQQ.1.1
- Or you can stop them ASLR
- This is how you do a picture:


Figure 1: Niap's Logo
- And this is how you reference it: Figure 1
- This is how you do an equation with an arbitrary counter:
$$x=\frac{-b\pm\sqrt{b^{2}-4ac}}{2a}$$ (1)
- And this is how you reference it: Eq. 1
- The following content should be included if:
  - "this" is selected from FQQ_QQQ.1.1

  Some text
- The following content should be included if:
  - the TOE implements "Widget Thing"

  Someting dependent on a feature
- And here's the audit event table for mandatory requirements.

RequirementAuditable EventsAdditional Audit Record Contents FQQ_QQQ.1 On failure of audit data capture due to

lack of disk space or pre-defined limit.

None. FQQ_QQQ.3No events specifiedFQQ_QQQ.2No events specifiedFQQ_QQQ.4No events specifiedFQQ_QQQ.6No events specifiedFMT_SMF.1/HOSTNo events specified FAU_GEN.1None.None. FAU_SAR.1None.None. FAU_STG.1None.None. FAU_STG_EXT.1 Failure of audit data capture due to lack of disk space or pre-defined limit.

On failure of logging function, capture record of failure and record upon restart of logging function. None. FAU_ARP.1No events specifiedFAU_SAA.1No events specified

1.

- Table 2 for more information.
- Test for an xref to section Section 3.1 Threats

And this is another sentence (or fragment). I added this sentence and deleted the next one. This uses the plural acronym OSes.

# 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]. |
| Base Protection Profile (Base-PP) | Protection Profile used as a basis to build a PP-Configuration. |
| Common Criteria (CC) | Common Criteria for Information Technology Security Evaluation (International Standard ISO/IEC 15408). |
| Common Criteria Testing Laboratory | Within the context of the Common Criteria Evaluation and Validation Scheme (CCEVS), an IT security evaluation facility, accredited by the National Voluntary Laboratory Accreditation Program (NVLAP) and approved by the NIAP Validation Body to conduct Common Criteria-based evaluations. |
| 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 (OE) | 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 (PP-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 the TOE. |
| 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 an ST. |
| Target of Evaluation (TOE) | The product under evaluation. |

### 1.2.2 Technical Terms

| | |
|---|---|
| Address Space Layout Randomization (ASLR) | An anti-exploitation feature which loads memory mappings into unpredictable locations. ASLR makes it more difficult for an attacker to redirect control to code that they have introduced into the address space of a process. |
| Administrator | An administrator is responsible for management activities, including setting policies that are applied by the enterprise on the operating system. This administrator could be acting remotely through a management server, from which the system receives configuration policies. An administrator can enforce settings on the system which cannot be overridden by non-administrator users. |
| Application (app) | Software that runs on a platform and performs tasks on behalf of the user or owner of the platform, as well as its supporting documentation. |
| Application Programming Interface (API) | A specification of routines, data structures, object classes, and variables that allows an application to make use of services provided by another software component, such as a library. APIs are often provided for a set of libraries included with the platform. |
| Credential | Data that establishes the identity of a user, e.g. a cryptographic key or password. |
| Critical Security Parameters (CSP) | Information that is either user or system defined and is used to operate a cryptographic module in processing encryption functions including cryptographic keys and authentication data, such as passwords, the disclosure or modification of which can compromise the security of a cryptographic module or the security of the information protected by the module. |
| DAR Protection | Countermeasures that prevent attackers, even those with physical access, from extracting data from non-volatile storage. Common techniques include data encryption and wiping. |
| Data Execution Prevention (DEP) | An anti-exploitation feature of modern operating systems executing on modern computer hardware, which enforces a non-execute permission on pages of memory. DEP prevents pages of memory from containing both data and instructions, which makes it more difficult for an attacker to introduce and execute code. |
| Developer | An entity that writes OS software. For the purposes of this document, vendors and developers are the same. |
| General Purpose Operating System | A class of OSes designed to support a wide-variety of workloads consisting of many concurrent applications or services. Typical characteristics for OSes in this class include support for third-party applications, support for multiple users, and security separation between users and their respective resources. General Purpose Operating Systems also lack the real-time constraint that defines Real Time Operating Systems (RTOS). RTOSes typically power routers, switches, and embedded devices. |
| Host-based Firewall | A software-based firewall implementation running on the OS for filtering inbound and outbound network traffic to and from processes running on the OS. |
| Operating System (OS) | Software that manages physical and logical resources and provides services for applications. The terms *TOE* and *OS* are interchangeable in this document. |
| Personally Identifiable Information (PII) | Any information about an individual maintained by an agency, including, but not limited to, education, financial transactions, medical history, and criminal or employment history and information which can be used to distinguish or trace an individual's identity, such as their name, social security number, date and place of birth, mother's maiden name, biometric records, etc., including any other personal information which is linked or linkable to an individual.[OMB] |
| Sensitive Data | Sensitive data may include all user or enterprise data or may be specific application data such as PII, emails, messaging, documents, calendar items, and contacts. Sensitive data must minimally include credentials and keys. Sensitive data shall be identified in the OS's TSS by the ST author. |
| User | A user is subject to configuration policies applied to the operating system by administrators. On some systems under certain configurations, a normal user can temporarily elevate privileges to that of an administrator. At that time, such a user should be considered an administrator. |
| Virtual Machine (VM) | Blah Blah Blah |

## 1.3 Compliant Targets of Evaluation

### 1.3.1 TOE Boundary

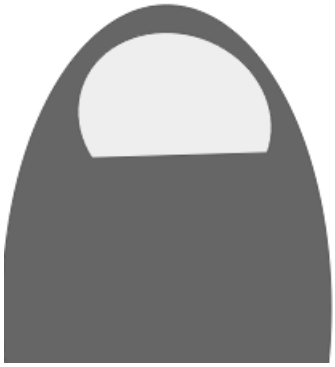


Figure 2: General TOE

### 1.3.2 TOE Platform

## 1.4 Use Cases

Requirements in this Protection Profile are designed to address the security problems in at least the following use cases. These use cases are intentionally very broad, as many specific use cases exist for an operating system. These use cases may also overlap with one another. An operating system's functionality may even be effectively extended by privileged applications installed onto it. However, these are out of scope of this PP.

[USE CASE 1] ~~End User Devices~~

Elephant-own device

This is everything we need to describe in words about this use case.

For a the list of appropriate selections and acceptable assignment values for this configuration, see C.1 Elephant-own device.

# 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~~ does not claim conformance to any packages

# 3 Security Problem Description

The security problem is described in terms of the threats that the OS is expected to address, assumptions about the operational environment, and any organizational security policies that the OS is expected to enforce.

## 3.1 Threats

T.NETWORK_ATTACK

An attacker is positioned on a communications channel or elsewhere on the network infrastructure. Attackers may engage in communications with applications and services running on or part of the OS with the intent of compromise. Engagement may consist of altering existing legitimate communications.

T.NETWORK_EAVESDROP

An attacker is positioned on a communications channel or elsewhere on the network infrastructure. Attackers may monitor and gain access to data exchanged between applications and services that are running on or part of the OS.

T.LOCAL_ATTACK

An attacker may compromise applications running on the OS. The compromised application may provide maliciously formatted input to the OS through a variety of channels including unprivileged system calls and messaging via the file system.

T.LIMITED_PHYSICAL_ACCESS

An attacker may attempt to access data on the OS while having a limited amount of time with the physical device.

## 3.2 Assumptions

A.PLATFORM
The OS relies upon a trustworthy computing platform for its execution. This underlying platform is out of scope of this PP.

A.PROPER_USER
The user of the OS is not willfully negligent or hostile, and uses the software in compliance with the applied enterprise security policy. At the same time, malicious software could act *as* the user, so requirements which confine malicious subjects are still in scope.

A.PROPER_ADMIN
The administrator of the OS is not careless, willfully negligent or hostile, and administers the OS within compliance of the applied enterprise security policy.

# 4 Security Objectives

## 4.1 Security Objectives for the TOE

O.ACCOUNTABILITY
Conformant OSes ensure that information exists that allows administrators to discover unintentional issues with the configuration and operation of the operating system and discover its cause. Gathering event information and immediately transmitting it to another system can also enable incident response in the event of system compromise.

O.INTEGRITY
Conformant OSes ensure the integrity of their update packages. OSes are seldom if ever shipped without errors, and the ability to deploy patches and updates with integrity is critical to enterprise network security. Conformant OSes provide execution environment-based mitigations that increase the cost to attackers by adding complexity to the task of compromising systems.

O.MANAGEMENT
To facilitate management by users and the enterprise, conformant OSes provide consistent and supported interfaces for their security-relevant configuration and maintenance. This includes the deployment of applications and application updates through the use of platform-supported deployment mechanisms and formats, as well as providing mechanisms for configuration and application execution control.

O.PROTECTED_STORAGE
To address the issue of loss of confidentiality of credentials in the event of loss of physical control of the storage medium, conformant OSes provide data-at-rest protection for credentials. Conformant OSes also provide access controls which allow users to keep their files private from other users of the same system.

O.PROTECTED_COMMS
To address both passive (eavesdropping) and active (packet modification) network attack threats, conformant OSes provide mechanisms to create trusted channels for CSP and sensitive data. Both CSP and sensitive data should not be exposed outside of the platform.

## 4.2 Security Objectives for the Operational Environment

The following security objectives for the operational environment assist the OS in correctly providing its security functionality. These track with the assumptions about the environment.

OE.PLATFORM
The OS relies on being installed on trusted hardware.

OE.PROPER_USER
The user of the OS is not willfully negligent or hostile, and uses the software within compliance of the applied enterprise security policy. Standard user accounts are provisioned in accordance with the least privilege model. Users requiring higher levels of access should have a separate account dedicated for that use.

OE.PROPER_ADMIN
The administrator of the OS is not careless, willfully negligent or hostile, and administers the OS within compliance of the applied enterprise security policy.

## 4.3 Security Objectives Rationale

This section describes how the assumptions, threats, and organization security policies map to the security objectives.

Table 1: Security Objectives Rationale

| Threat, Assumption, or OSP | Security Objectives | Rationale |
|---|---|---|
| T.NETWORK_ATTACK | O.PROTECTED_COMMS | The threat T.NETWORK_ATTACK is countered by O.PROTECTED_COMMS as this provides for integrity of transmitted data. |
| | O.INTEGRITY | The threat T.NETWORK_ATTACK is countered by O.INTEGRITY as this provides for integrity of software that is installed onto the system from the network. |
| | O.MANAGEMENT | The threat T.NETWORK_ATTACK is countered by O.MANAGEMENT as this provides for the ability to configure the OS to defend against network attack. |
| | | The threat T.NETWORK_ATTACK is countered by |

| | | |
|---|---|---|
| | O.ACCOUNTABILITY | O.ACCOUNTABILITY as this provides a mechanism for the OS to report behavior that may indicate a network attack has occurred. |
| | O.PROTECTED_COMMS | The threat T.NETWORK_EAVESDROP is countered by O.PROTECTED_COMMS as this provides for confidentiality of transmitted data. |
| T.NETWORK_EAVESDROP | | |
| | O.MANAGEMENT | The threat T.NETWORK_EAVESDROP is countered by O.MANAGEMENT as this provides for the ability to configure the OS to protect the confidentiality of its transmitted data. |
| | O.INTEGRITY | The objective O.INTEGRITY protects against the use of mechanisms that weaken the TOE with regard to attack by other software on the platform. |
| T.LOCAL_ATTACK | | |
| | O.ACCOUNTABILITY | The objective O.ACCOUNTABILITY protects against local attacks by providing a mechanism to report behavior that may indicate a local attack is occurring or has occurred. |
| T.LIMITED_PHYSICAL_ACCESS | O.PROTECTED_STORAGE | The objective O.PROTECTED_STORAGE protects against unauthorized attempts to access physical storage used by the TOE. |
| A.PLATFORM | OE.PLATFORM | The operational environment objective OE.PLATFORM is realized through A.PLATFORM. |
| A.PROPER_USER | OE.PROPER_USER | The operational environment objective OE.PROPER_USER is realized through A.PROPER_USER. |
| A.PROPER_ADMIN | OE.PROPER_ADMIN | The operational environment objective OE.PROPER_ADMIN is realized through A.PROPER_ADMIN. |

# 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 conventions are used for the completion of operations:

- **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."

This chapter describes the security requirements which have to be fulfilled by the OS. 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**): is used to add details to a requirement, 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: are identified with a number inside parentheses (e.g. "(1)")

## 5.1 Security Functional Requirements

### 5.1.1 Something

Here's whwere we talk about an audit table.

Table 2: Audit Events for Mandatory Requirements

| Requirement | Auditable Events | Additional Audit Record Contents |
|---|---|---|
| FQQ_QQQ.1 | On failure of audit data capture due to lack of disk space or pre-defined limit. | None. |
| | Failure of audit data capture due to lack of disk space or pre-defined limit. On failure | |

| | | |
|---|---|---|
| FAU_STG_EXT.1 | of logging function, capture record of failure and record upon restart of logging function. | |

### 5.1.2 QQQQ

**FOO_FOO.1 Foo Foo**

FOO_FOO.1.1
The TOE shall consiste of |**selection**: *soup*, *salad*] followed by |**selection**: *pizza*, *spaghetti*, *ratatouille*, *sushi*] with |**selection**: *white*, *red*].
Application Note:
Evaluation Activity
*Something*
*TSS*
*ABC*

**FOO_BAR.1 Foo Bar**

FOO_BAR.1.1
The TOE shall drink [**selection**: *tea*, *coffee*].
Application Note:
Evaluation Activity
*Something*
*TSS*
*ABC*
FOO_BAR.1.2
The TOE shall eat |**selection**: *crackers*, *nothing*]
Application Note: Deonstrating rules across elements
Evaluation Activity
*Something*
*Guidance*
*Some guidance*

### 5.1.3 QQQQ

**FQQ_QQQ.1 QQQQQ**

FQQ_QQQ.1.1
The TOE shall do either [**selection**: *this*, *that*].
Application Note:
Evaluation Activity
*TSS*
*Activities assoiated with the TSS.*
*Guidance*
*Activities assoiated with guidance*
*Tests*

- ***Test 1:*** *Make shadow puppets.*
  ***Objective:****This is the motivation behind the tests.*
  ***Evidence:****A warm fuzzy feeling*

*Activities assoiated with the Tests.*
*The following content should be included if:*

- *For virtual TOEs*

*Great tests for something virtual.*
*The following content should be included if:*

- *For physical/imaginary TOEs*

*Great tests for something tangible or in my mind.*

### 5.1.~~2~~ 4 Security Management (FMT)

**FMT_SMF.1/HOST Specification of Management Functions (EDR Management of Host Agent)**

FMT_SMF.1.1/HOST
The EDR shall be capable of performing the following functions that control behavior of the Host Agent:
~~XX~~

| # | Management Function | Administrator | SOC Analyst | Read-Only User |
|---|---|---|---|---|
| 1 | Configure the time frame for sending Host Agent data to the EDR [**assignment**: list of configurable time frames] | M | O | - |
| 2 | Assign a label or tag to categorize or group individual endpoint systems | M | O | - |

Application Note: This requirement captures all the configuration functionality the EDR provides the administrator to configure the EDR Host Agents.

Chart legend: X = Mandatory, O = Optional, - = N/A
Evaluation Activity
*TSS*
*The evaluator shall verify the ST contains a list of roles and what functions they can perform. The evaluator shall verify the list matches the chart in the requirement.*
*Guidance*
*The evaluator shall review the operational guidance to verify that the EDR has documented capabilities to perform the management functions.*
*Tests*
*The evaluator shall perform the below tests:*

- ***Test 1:** The evaluator shall modify the time frame for sending Host Agent data to the EDR and verify that an affected Host Agent is sending data at the intended interval.*
- ***Test 2:** The evaluator shall tag or categorize a group of individual endpoint systems and verify that the tag or categorization persists within the EDR management dashboard for other users.*
- ***Test 3:** The evaluator shall attempt each function with each role and verify access conforms with the chart in the requirement.*

***Objective:**This is the motivation behind the tests.*
***Evidence:**A check should appear.*

## 5.1.~~3~~ 5 Security Audit (FAU)

### FAU_GEN.1 Audit Data Generation

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

a. Start-up and shutdown of audit functions
b. All administrative actions
c. [Specifically defined auditable events in Table 1]
d. [**selection**: *additional information defined in Table 2*, *additional information defined in Table 3*, *additional information defined in Table 4*, *additional information defined in in Table 5*, *no other information*]

FAU_GEN.1.2
The TSF 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. [**selection**: *additional information defined in Table 2*, *additional information defined in Table 3*, *additional information defined in Table 4*, *additional information defined in in Table 5*, *no other information*]

Application Note: The ST author can include other auditable events directly in Table 1; they are not limited to the list presented. The ST 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 VM, 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 ST if the associated SFRs and selections are included.

The Table 1 entry for FDP_VNC_EXT.1 refers to configuration settings that attach ~~VMs~~ VMs to virtualized network components. Changes to these configurations can be made during VM execution or when ~~VMs~~ VMs 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 VM is connected to a physical device (when the device becomes part of the VM’s hardware view), not to log every time that the device is accessed. Generally, this is only once at VM 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 TSS 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 TSS.*

*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 TOE that are necessary to enforce the requirements specified in thePP. 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 TOE's ability to correctly generate audit records by having theTOE 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 ~~1: Auditable Events~~ 2 for more information.* ~~*Table 1: Auditable Events*~~ ~~Requirement~~~~Auditable Events~~~~Additional Audit Record Contents~~ ~~FQQ_QQQ.1~~ ~~On failure of audit data capture due to lack of disk space or pre-defined limi~~t.

~~None.~~ ~~FQQ_QQQ.3~~~~No events specified~~~~FQQ_QQQ.2~~~~No events specified~~~~FQQ_QQQ.4~~~~No events specified~~~~FQQ_QQQ.6~~~~No events specified~~~~FMT_SMF.1/HOST~~~~No events specified~~

**FAU_**

~~GEN.1~~~~None.~~~~None.~~~~FAU_~~

**SAR.1**

~~None.~~~~None.~~ ~~FAU_STG.1~~~~None.~~~~None.~~ ~~FAU_STG_EXT.1~~ ~~Failure of audit data capture due to lack of disk space or pre-defined limit.~~

~~On failure of logging function, capture record of failure and record upon restart of logging function. None.~~ ~~FAU_ARP.1~~~~No events specified~~~~FAU_SAA.1~~~~No events specified~~ ~~The following audit events are included if:~~

- ~~"additional information defined in Table 2" is selected from~~ ~~FAU_GEN.1.1~~

~~Table 2: Auditable Events~~ ~~Requirement~~~~Auditable Events~~~~Additional~~

**Audit**

~~Record Contents~~ ~~FQQ_QQQ.1~~ ~~On number of turtles in the area.None.~~ ~~FQQ_QQQ.3~~~~No events specified~~~~FQQ_QQQ.2~~~~No events specified~~~~FQQ_QQQ.4~~~~No events specified~~~~FQQ_QQQ.6~~~~No events specified~~~~FMT_SMF.1/HOST~~~~No events specified~~~~FAU_GEN.1~~~~No events specified~~~~FAU_SAR.1~~~~No events specified~~~~FAU_STG.1~~~~No events specified~~~~FAU_STG_EXT.1~~~~No events specified~~~~FAU_ARP.1~~~~No events specified~~~~FAU_SAA.1~~~~No events specified~~

**~~FAU_SAR.1 Audit~~ Review**

FAU_SAR.1.1
The TSF shall provide [administrators] with the capability to read [all information] from the audit records.
FAU_SAR.1.2
The TSF shall provide the audit records in a manner suitable for the user to interpret the information.
Evaluation Activity
*Guidance*
*The evaluator shall review the operational guidance for the procedure on how to review the audit records.*
*Tests*
*The evaluator shall verify that the audit records provide all of the information specified in FAU_GEN.1 and that this information is suitable for human interpretation. The assurance activity for this requirement is performed in conjunction with the assurance activity for FAU_GEN.1.*

**FAU_STG.1 Protected Audit Trail Storage**

FAU_STG.1.1
The TSF shall protect the stored audit records in the audit trail from unauthorized deletion.
FAU_STG.1.2
The TSF shall be able to [prevent] modifications to the stored audit records in the audit trail.

Application Note: The assurance activity for this SFR is not intended to imply that the TOE must support an administrator's ability to designate individual audit records for deletion. That level of granularity is not required.
Evaluation Activity
*The evaluator shall ensure that the TSS describes how the audit records are protected from unauthorized modification or deletion. The evaluator shall ensure that the TSS describes the conditions that must be met for authorized deletion of audit records. The evaluator shall perform the following tests:*

*Tests*

- ***Test 1:*** *The evaluator shall access the audit trail as an unauthorized Administrator and attempt to modify and delete the audit records. The evaluator shall verify that these attempts fail.*
- ***Test 2:*** *The evaluator shall access the audit trail as an authorized Administrator and attempt to delete the audit records. The evaluator shall verify that these attempts succeed. The evaluator shall verify that only the records authorized for deletion are deleted.*

**FAU_STG_EXT.1 Off-Loading of Audit Data**

FAU_STG_EXT.1.1
The TSF shall be able to transmit the generated audit data to an external IT entity using a trusted channel as specified in FTP_ITC_EXT.1.
Evaluation Activity
*Protocols used for implementing the trusted channel must be selected in FTP_ITC_EXT.1.*
*TSS*
*The evaluator shall examine the TSS to ensure it describes the means by which the audit data are transferred to the external audit server, and how the trusted channel is provided.*
*Guidance*
*The evaluator shall examine the operational guidance to ensure it describes how to establish the trusted channel to the audit server, as well as describe any requirements on the audit server (particular audit server protocol, version of the protocol required, etc.), as well as configuration of the TOE needed to communicate with the audit server.*
*Tests*
*Testing of the trusted channel mechanism is to be performed as specified in the assurance activities for FTP_ITC_EXT.1.*

*The evaluator shall perform the following test for this requirement:*

- ***Test 1:*** *The evaluator shall establish a session between the TOE and the audit server according to the configuration guidance provided. The evaluator shall then examine the traffic that passes between the audit server and the TOE during several activities of the evaluator's choice designed to generate audit data to be transferred to the audit server. The evaluator shall observe that these data are not able to be viewed in the clear during this transfer, and that they are successfully received by the audit server. The evaluator shall record the particular software (name, version) used on the audit server during testing.*

FAU_STG_EXT.1.2
The TSF shall [**selection**: *drop new audit data*, *overwrite previous audit records according to the following rule*: [***assignment***: *rule for overwriting previous audit records*], [***assignment***: *other action*]] when the local storage space for audit data is full.
Application Note: An external log server, if available, might be used as alternative storage space in case the local storage space is full. An 'other action' could be defined in this case as 'send the new audit data to an external IT entity'.
Evaluation Activity
*TSS*
*The evaluator shall examine the TSS to ensure it describes what happens when the local audit data store is full.*
*Guidance*
*The evaluator shall also examine the operational guidance to determine that it describes the relationship between the local audit data and the audit data that are sent to the audit log server. For example, when an audit event is generated, is it simultaneously sent to the external server and the local store, or is the local store used as a buffer and "cleared" periodically by sending the data to the audit server.*
*Tests*
*The evaluator shall perform operations that generate audit data and verify that this data is stored locally. The evaluator shall perform operations that generate audit data until the local storage space is exceeded and verifies that the TOE complies with the behavior defined in the ST for FAU_STG_EXT.1.2.*

## 5.1.4~~-6~~ TOE Security Functional Requirements Rationale

The following rationale provides justification for each security objective for the TOE, showing that the SFRs are suitable to meet and achieve the security objectives:

Table 3: SFR Rationale

| OBJECTIVE | ADDRESSED BY | RATIONALE |
|---|---|---|
| O.ACCOUNTABILITY | FAU_GEN.1<br>FTP_ITC_EXT.1 | 'cause FAU_GEN.1 is awesome<br>Cause FTP |

| | | |
|---|---|---|
| | | reasons |
| | FPT_SBOP_EXT.1 | For reasons |
| | FPT_ASLR_EXT.1 | ASLR For reasons |
| | FPT_TUD_EXT.1 | For reasons |
| | FPT_TUD_EXT.2 | For reasons |
| | FCS_COP.1/HASH | For reasons |
| | FCS_COP.1/SIGN | For reasons |
| O.INTEGRITY | FCS_COP.1/KEYHMAC | For reasons |
| | FPT_ACF_EXT.1 | For reasons |
| | FPT_SRP_EXT.1 | For reasons |
| | FIA_X509_EXT.1 | For reasons |
| | FPT_TST_EXT.1 | For reasons |
| | FTP_ITC_EXT.1 | For reasons |
| | FPT_W^X_EXT.1 | For reasons |
| | FIA_AFL.1 | For reasons |
| | FIA_UAU.5 | For reasons |
| | FMT_MOF_EXT.1 | For reasons |
| O.MANAGEMENT | FMT_SMF_EXT.1 | For reasons |
| | FTA_TAB.1 | For reasons |
| | FTP_TRP.1 | For reasons |
| O.PROTECTED_STORAGE | FCS_STO_EXT.1, FCS_RBG_EXT.1, FCS_COP.1/ENCRYPT, FDP_ACF_EXT.1 | Rationale for a big chunk |
| O.PROTECTED_COMMS | FCS_RBG_EXT.1, FCS_CKM.1, FCS_CKM.2, FCS_CKM_EXT.4, FCS_COP.1/ENCRYPT, FCS_COP.1/HASH, FCS_COP.1/SIGN, FCS_COP.1/HMAC, FDP_IFC_EXT.1, FIA_X509_EXT.1, FIA_X509_EXT.2, FTP_ITC_EXT.1 | Rationale for a big chunk |

## 5.2 Security Assurance Requirements

The Security Objectives in Section 4 Security Objectives were constructed to address threats identified in Section 3.1 Threats. The Security Functional Requirements (SFRsSFRs) in Section 5.1 Security Functional Requirements are a formal instantiation of the Security Objectives. The PP identifies the Security Assurance Requirements (SARsSARs) to frame the extent to which the evaluator assesses the documentation applicable for the evaluation and performs independent testing. This section lists the set of SARs SARs from CC part 3 that are required in evaluations against this PP. Individual Assurance Activities to o be performed are specified both in Section 5.1 Security Functional Requirements as well as in this section. The general model for evaluation of OSs OSs against STs written to conform to this PP is as follows: After the ST has been approved for evaluation, the ITSEF TSEF will obtain the OS, supporting environmental IT, and the administrative/user guides for the OS. The ITSEF is expected to perform actions mandated by the Common Evaluation Methodology (CEM) for the ASE and ALC SARs. The ITSEF also performs the Assurance Activities contained within Section 5.1 Security Functional Requirements, which are intended to be an interpretation of the other CEM assurance requirements as they apply to the specific technology instantiated in the OS. The Assurance Activities that are captured in Section 5.1 Security Functional Requirements also provide clarification as to what the developer needs to provide to demonstrate the OS is compliant with the PP.

### 5.2.1 Class ASE: Security Target

As per ASE activities defined in [CEM].

### 5.2.2 Class ADV: Development

The information about the OS is contained in the guidance documentation available to the end user as well as the TSS portion of the ST. The OS developer must concur with the description of the product that is contained in the TSS as it relates to the functional requirements. The Assurance Activities contained in Section 5.1 Security Functional Requirements should provide the ST authors with sufficient information to determine the appropriate content for the TSS section.

**ADV_FSP.1 Basic Functional Specification (ADV_FSP.1)**

The functional specification describes the TSFIs. It is not necessary to have a formal or complete specification of these interfaces. Additionally, because OSs conforming to this PP will necessarily have interfaces to the Operational Environment that are not directly invokable by OS users, there is little point specifying that such interfaces be described in and of themselves since only indirect testing of such interfaces may be possible. For this PP, the activities for this family should focus on understanding the interfaces presented in the TSS in response to the functional requirements and the interfaces presented in the AGD documentation. No additional “functional specification” documentation is necessary to satisfy the assurance activities specified. The interfaces that need to be evaluated are characterized through the information needed to perform the assurance activities listed, rather than as an independent, abstract list.

**Developer action elements:**

ADV_FSP.1.1D
The developer shall provide a functional specification.

**Content and presentation elements:**

ADV_FSP.1.2C
The developer shall provide a tracing from the functional specification to the SFRs.
Application Note: As indicated in the introduction to this section, the functional specification is comprised of the information contained in the AGD_OPE and AGD_PRE documentation. The developer may reference a website accessible to application developers and the evaluator. The assurance activities in the functional requirements point to evidence that should exist in the documentation and TSS 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.
ADV_FSP.1.3C
The functional specification shall describe the purpose and method of use for each SFR-enforcing and SFR-supporting TSFI.
ADV_FSP.1.4C
The functional specification shall identify all parameters associated with each SFR-enforcing and SFR-supporting TSFI.
ADV_FSP.1.5C
The functional specification shall provide rationale for the implicit categorization of interfaces as SFR-non-interfering.
ADV_FSP.1.6C
The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

**Evaluator action elements:**

ADV_FSP.1.7E
The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_FSP.1.8E
The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.
Evaluation Activity
*There are no specific assurance activities associated with these SARs, except ensuring the information is provided. The functional specification documentation is provided to support the evaluation activities described in Section 5.1 Security Functional Requirements, 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 there is insufficient interface information, then an adequate functional specification has not been provided.*

### 5.2.3 Class AGD: Guidance Documentation

The guidance documents will be provided with the ST. Guidance must include a description of how theIT personnel verifies that the Operational Environment can fulfill its role for the security functionality. The documentation should be in an informal style and readable by the IT personnel. Guidance must be provided for every operational environment that the product supports as claimed in the ST. This guidance includes instructions to successfully install theTSF in that environment; and Instructions to manage the security of the TSF as a product and as a component of the larger operational environment. Guidance pertaining to particular security functionality is also provided; requirements on such guidance are contained in the assurance activities specified with each requirement.

**AGD_OPE.1 Operational User Guidance (AGD_OPE.1)**

**Developer action elements:**

AGD_OPE.1.1D
The developer shall provide operational user guidance.
Application Note: The operational user guidance does not have to be contained in a single document. Guidance to users, administrators and application developers can be spread among documents or web pages. Rather than repeat information here, the developer should review the assurance activities for this component to ascertain the specifics of the guidance that the evaluator will be checking for. This will provide the necessary information for the preparation of acceptable guidance.

**Content and presentation elements:**

AGD_OPE.1.2C
The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.
Application Note: User and administrator are to be considered in the definition of user role.
AGD_OPE.1.3C
The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the OS in a secure manner.
AGD_OPE.1.4C
The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.

Application Note: This portion of the operational user guidance should be presented in the form of a checklist that can be quickly executed by IT personnel (or end-users, when necessary) and suitable for use in compliance activities. When possible, this guidance is to be expressed in the eXtensible Configuration Checklist Description Format (XCCDF) to support security automation. Minimally, it should be presented in a structured format which includes a title for each configuration item, instructions for achieving the secure configuration, and any relevant rationale.
AGD_OPE.1.5C
The operational user guidance shall, for each user role, 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 TSF.
AGD_OPE.1.6C
The operational user guidance shall identify all possible modes of operation of the OS (including operation following failure or operational error), their consequences, and implications for maintaining secure operation.
AGD_OPE.1.7C
The operational user guidance shall, for each user role, describe the security measures to be followed in order to fulfill the security objectives for the operational environment as described in the ST.
AGD_OPE.1.8C
The operational user guidance shall be clear and reasonable.

**Evaluator action elements:**

AGD_OPE.1.9E
The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
Evaluation Activity
*Some of the contents of the operational guidance are verified by the assurance activities in Section 5.1 Security Functional Requirements and evaluation of the OS according to the [CEM]. The following additional information is also required. If cryptographic functions are provided by the OS, the operational guidance shall contain instructions for configuring the cryptographic engine associated with the evaluated configuration of the OS. It shall provide a warning to the administrator that use of other cryptographic engines was not evaluated nor tested during the CC evaluation of the OS. The documentation must describe the process for verifying updates to the OS by verifying a digital signature – this may be done by the OS or the underlying platform. The evaluator will verify that this process includes the following steps: Instructions for obtaining the update itself. This should include instructions for making the update accessible to the OS (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 OS will likely contain security functionality that does not fall in the scope of evaluation under this PP. The operational guidance shall make it clear to an administrator which security functionality is covered by the evaluation activities.*

**AGD_PRE.1 Preparative Procedures (AGD_PRE.1)**

**Developer action elements:**

AGD_PRE.1.1D
The developer shall provide the OS, including its preparative procedures.
Application 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.1.2C
The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered OS in accordance with the developer's delivery procedures.
AGD_PRE.1.3C
The preparative procedures shall describe all the steps necessary for secure installation of the OS and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.

**Evaluator action elements:**

AGD_PRE.1.4E
The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
AGD_PRE.1.5E
The evaluator shall apply the preparative procedures to confirm that the OS can be prepared securely for operation.
Evaluation Activity
*As indicated in the introduction above, there are significant expectations with respect to the documentation—especially when configuring the operational environment to support OS functional requirements. The evaluator shall check to ensure that the guidance provided for the OS adequately addresses all platforms claimed for the OS in the ST.*

## 5.2.4 Class ALC: Life-cycle Support

At the assurance level provided for OSs conformant to this PP, life-cycle support is limited to end-user-visible aspects of the life-cycle, rather than an examination of the OS vendor’s development and configuration management process. This is not meant to diminish the critical role that a developer’s practices play in contributing to the overall trustworthiness of a product;

rather, it is a reflection on the information to be made available for evaluation at this assurance level.

**ALC_CMC.1 Labeling of the TOE (ALC_CMC.1)**

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

**Developer action elements:**

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

**Content and presentation elements:**

ALC_CMC.1.2C
The OS shall be labeled with a unique reference.
Application Note: Unique reference information includes:

- OS Name
- OS Version
- OS Description
- Software Identification (SWID) tags, if available

**Evaluator action elements:**

ALC_CMC.1.3E
The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
Evaluation Activity
*The evaluator will 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. Further, the evaluator will check the AGD guidance and OS 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 OS, the evaluator will 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 (ALC_CMS.1)**

Given the scope of the OS and its associated evaluation evidence requirements, this component's assurance activities are covered by the assurance activities listed for ALC_CMC.1.

**Developer action elements:**

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

**Content and presentation elements:**

ALC_CMS.1.2C
The configuration list shall include the following: the OS itself; and the evaluation evidence required by the SARs.
ALC_CMS.1.3C
The configuration list shall uniquely identify the configuration items.

**Evaluator action elements:**

ALC_CMS.1.4E
The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
Evaluation Activity
*The "evaluation evidence required by the SARs" in this PP is limited to the information in the ST coupled with the guidance provided to administrators and users under the AGD requirements. By ensuring that the OS is specifically identified and that this identification is consistent in the ST and in the AGD guidance (as done in the assurance activity for ALC_CMC.1), the evaluator implicitly confirms the information required by this component. Life-cycle support is targeted aspects of the developer's life-cycle and instructions to providers of applications for the developer's devices, rather than an in-depth examination of the TSF manufacturer's development and configuration management process. This is not meant to diminish the critical role that a developer's practices play in contributing to the overall trustworthiness of a product; rather, it's a reflection on the information to be made available for evaluation.*
*The evaluator will ensure that the developer has identified (in guidance documentation for application developers concerning the targeted 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 will ensure that this documentation also includes an indication of whether such protections are on by default, or have to be specifically enabled. The evaluator will 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**

This component requires the OS developer, in conjunction with any other necessary parties, to provide information as to how the end-user devices are updated to address security issues in a timely manner. The documentation describes the process of providing updates to the public from the time a security flaw is reported/discovered, to the time an update is released. This description includes the parties involved (e.g., the developer, carriers(s)) and the steps that are performed (e.g., developer testing, carrier testing), including worst case time periods, before an update is made available to the public.

**Developer action elements:**

ALC_TSU_EXT.1.1D
The developer shall provide a description in the TSS of how timely security updates are made to the OS.
ALC_TSU_EXT.1.2D
The developer shall provide a description in the TSS of how users are notified when updates change security properties or the configuration of the product.

**Content and presentation elements:**

ALC_TSU_EXT.1.3C
The description shall include the process for creating and deploying security updates for the OS software.
ALC_TSU_EXT.1.4C
The description shall include the mechanisms publicly available for reporting security issues pertaining to the OS.
Note: The reporting mechanism could include web sites, email addresses, as well as 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.1.5E
The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
Evaluation Activity
*The evaluator will verify that the TSS contains a description of the timely security update process used by the developer to create and deploy security updates. The evaluator will verify that this description addresses the entire application. The evaluator will also verify that, in addition to the OS developer's process, any third-party processes are also addressed in the description. The evaluator will also verify that each mechanism for deployment of security updates is described.*
*The evaluator will verify that, for each deployment mechanism described for the update process, the TSS lists a time between public disclosure of a vulnerability and public availability of the security update to the OS patching this vulnerability, to include any third-party or carrier delays in deployment. The evaluator will verify that this time is expressed in a number or range of days.*
*The evaluator will verify that this description includes the publicly available mechanisms (including either an email address or website) for reporting security issues related to the OS. The evaluator shall verify that the description of this mechanism includes a method for protecting the report either using a public key for encrypting email or a trusted channel for a website.*

### 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 the 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 – Conformance (ATE_IND.1)**

Testing is performed to confirm the functionality described in the TSS as well as the administrative (including configuration and operational) documentation provided. The focus of the testing is to confirm that the requirements specified in Section 5.1 Security Functional Requirements being met, although some additional testing is specified for SARs in Section 5.2 Security Assurance Requirements. The Assurance Activities identify the additional testing activities associated with these components. The evaluator produces a test report documenting the plan for and results of testing, as well as coverage arguments focused on the platform/OS combinations that are claiming conformance to this PP. Given the scope of the OS and its associated evaluation evidence requirements, this component's assurance activities are covered by the assurance activities listed for ALC_CMC.1.

**Developer action elements:**

ATE_IND.1.1D
The developer shall provide the OS for testing.

**Content and presentation elements:**

ATE_IND.1.2C
The OS shall be suitable for testing.

**Evaluator action elements:**

ATE_IND.1.3E
The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.
ATE_IND.1.4E
The evaluator shall test a subset of the TSF to confirm that the TSF operates as specified.
Application Note: The evaluator will test the OS on the most current fully patched version of the platform.
Evaluation Activity
*The evaluator will prepare a test plan and report documenting the testing aspects of the system, including any application crashes during testing. The evaluator shall determine the root cause of any application crashes and include that information in the report. The test plan covers all of the testing actions contained in the [CEM] and the body of this PP's Assurance Activities.*
*While it is not necessary to have one test case per test listed in an Assurance Activity, the evaluator must document in the test plan that each applicable testing requirement in the ST 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 ST, 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 ST 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 evaluator is 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) should be provided that the driver or tool will not adversely affect the performance of the functionality by the OS and its platform.*
*This also includes the configuration of the cryptographic engine to be used. The cryptographic algorithms implemented by this engine are those specified by this PP and used by the cryptographic protocols being evaluated (IPsec, TLS). 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 discover 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, the evaluator will not be expected to test for these vulnerabilities in the OS. 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 protection profiles.

**AVA_VAN.1 Vulnerability Survey (AVA_VAN.1)**

**Developer action elements:**

AVA_VAN.1.1D
The developer shall provide the OS for testing.

**Content and presentation elements:**

AVA_VAN.1.2C
The OS shall be suitable for testing.

**Evaluator action elements:**

AVA_VAN.1.3E
The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
AVA_VAN.1.4E
The evaluator shall perform a search of public domain sources to identify potential vulnerabilities in the OS.
Application Note: Public domain sources include the Common Vulnerabilities and Exposures (CVE) dictionary for publicly-known vulnerabilities. Public domain sources also include sites which provide free checking of files for viruses.
AVA_VAN.1.5E
The evaluator shall conduct penetration testing, based on the identified potential vulnerabilities, to determine that the OS is resistant to attacks performed by an attacker possessing Basic attack potential.
Evaluation Activity
*The evaluator will 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 find vulnerabilities that have been found in similar applications with a particular focus on network protocols the*

*application uses and document formats it parses. 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. 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 the TOE) are contained in the body of this PP. This ~~Appendix~~ 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 the TOE 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 this PP.

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~~ based Requirements ) are dependent on the TOE implementing a particular function. If the TOE fulfills any of these requirements, the vendor must either add the related SFR or disable the functionality for the evaluated configuration.

## A.1 Strictly Optional Requirements

### A.1.1 QQQQ

#### FQQ_QQQ.2 TQQQQQ

FQQ_QQQ.2.1
The TOE shall do soemthing.
Application Note:
Evaluation Activity
*TSS*
*Activities assoiated with the TSS.*

### A.1.2 Security Audit (FAU)

#### FAU_ARP.1 Security Audit Automatic Response

FAU_ARP.1.1
The TSF shall take [**assignment**: list of actions] upon detection of a potential security violation.
Application Note: In certain cases, it may be useful for Virtualization Systems to perform automated responses to certain security events. An example may include halting a VM which has taken some action to violate a key system security policy. This may be especially useful with headless endpoints when there is no human user in the loop.

The potential security violation mentioned in FAU_ARP.1.1 refers to FAU_SAA.1.
Evaluation Activity
*Tests*
*The evaluator shall generate a potential security violation as defined in FAU_SAA.1 and verify that each action in the assignment in FAU_ARP.1.1 is performed by the TSF as a result. The evaluator shall perform this action for each security violation that is defined in FAU_SAA.1.*

#### FAU_SAA.1 Security Audit Analysis

FAU_SAA.1.1
The TSF shall be able to apply a set of rules in monitoring the audited events and based upon these rules indicate a potential violation of the enforcement of the SFRs.
FAU_SAA.1.2
The TSF shall enforce the following rules for monitoring audited events:

a. accumulation or combination of [**assignment**: subset of defined auditable events] known to indicate a potential security violation
b. [**assignment**: any other rules]

Application Note: The potential security violation described in FAU_SAA.1 can be used as a trigger for automated responses as defined in FAU_ARP.1.
Evaluation Activity
*Tests*
*The evaluator shall cause each combination of auditable events defined in FAU_SAA.1.2 to occur, and verify that a potential security violation is indicated by the TSF.*

## A.2 Objective Requirements

### A.2.1 QQQQ

#### FQQ_QQQ.3 BQQQQQ

FQQ_QQQ.3.1
The TOE shall do [**assignment**: guidance on what things should be assignable].
Application Note: Notes. Notes. Notes.
Evaluation Activity
*Guidance*
*Activities assoiated with guidance*

## A.3 Implementation-

~~Dependent~~

## based Requirements

### A.3.1 Widget Thing

This is a super description of this certain feature.

If this is implemented by the TOE

~~includes the widget thing~~

~~all of~~

the following

~~SFRs~~

requirements must be

~~claimed~~

included in the ST;

- **FQQ_QQQ.6**

### A.3.1.1 QQQQ

#### FQQ_QQQ.6 WQQQQQ

FQQ_QQQ.6.1
The TOE shall do something with regards to some implementation.
Application Note:
Evaluation Activity
*TSS*
*Activities assoiated with the TSS.*

# Appendix B - Selection-~~Based~~ based Requirements

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

## B.1 QQQQ

**FQQ_QQQ.4 UQQQQQ**

***~~This is a~~ The inclusion of this selection-based component ~~. Its inclusion~~ depends upon a selection ~~from~~ in FQQ_QQQ.1.1.***
FQQ_QQQ.4.1
The TOE shall do something great.
Application Note:
Evaluation Activity
*TSS*
*Activities assoiated with the TSS.*

# Appendix C - Use Case Templates

## C.1 Elephant-own device

From FOO_FOO.1.1:
* select soup
Choose something other than:
* ratatouille
* pizza
From FAU_GEN.1.1:
Choose something other than:
* additional information defined in Table 2
Include FQQ_QQQ.2 in the ST

# Appendix D - Inherently Satisfied Requirements

This appendix lists requirements that should be considered satisfied by products successfully evaluated against this Protection Profile. However, these requirements are not featured explicitly as SFRs and should not be included in the ST. They are not included as standalone SFRs because it would increase the time, cost, and complexity of evaluation. This approach is permitted by [CC] Part 1, **8.2 Dependencies between components**.
This information benefits systems engineering activities which call for inclusion of particular security controls. Evaluation against the Protection Profile provides evidence that these controls are present and have been evaluated.

| Requirement | Rationale for Satisfaction |
|---|---|
| FIA_UAU.1 - Timing of authentication | FIA_AFL.1 implicitly requires that the OS perform all necessary actions, including those on behalf of the user who has not been authenticated, in order to authenticate; therefore it is duplicative to include these actions as a separate assignment and test. |
| FIA_UID.1 - Timing of identification | FIA_AFL.1 implicitly requires that the OS perform all necessary actions, including those on behalf of the user who has not been identified, in order to authenticate; therefore it is duplicative to include these actions as a separate assignment and test. |
| FMT_SMR.1 - Security roles | FMT_MOF_EXT.1 specifies role-based management functions that implicitly defines user and privileged accounts; therefore, it is duplicative to include separate role requirements. |
| FPT_STM.1 - Reliable time stamps | FAU_GEN.1.2 explicitly requires that the OS associate timestamps with audit records; therefore it is duplicative to include a separate timestamp requirement. |
| FTA_SSL.1 - TSF-initiated session locking | FMT_MOF_EXT.1 defines requirements for managing session locking; therefore, it is duplicative to include a separate session locking requirement. |
| FTA_SSL.2 - User-initiated locking | FMT_MOF_EXT.1 defines requirements for user-initiated session locking; therefore, it is duplicative to include a separate session locking requirement. |
| FAU_STG.1 - Protected audit trail storage | FPT_ACF_EXT.1 defines a requirement to protect audit logs; therefore, it is duplicative to include a separate protection of audit trail requirements. |
| FAU_GEN.2 - User identity association | FAU_GEN.1.2 explicitly requires that the OS record any user account associated with each event; therefore, it is duplicative to include a separate requirement to associate a user account with each event. |
| FAU_SAR.1 - Audit review | FPT_ACF_EXT.1.2 requires that audit logs (and other objects) are protected from reading by unprivileged users; therefore, it is duplicative to include a separate requirement to protect only the audit information. |

# Appendix

~~D - Entropy Documentation and Assessment This appendix describes the required supplementary information for the entropy source used by the OS.~~

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

## D.1 Design Description

Documentation shall include the design of the entropy source as a whole, including the interaction of all entropy source components. Any information that can be shared regarding the design should also be included for any third-party entropy sources that are included in the product.

The documentation will describe the operation of the entropy source to include, 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 entropy comes from, where the entropy output is passed next, any post-processing of the raw outputs (hash, XOR, etc.), if/where it is stored, and finally, how it is output from the entropy source. Any conditions placed on the process (e.g., blocking) should also be described in the entropy source design. Diagrams and examples are encouraged.

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

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

## D.2 Entropy Justification

There should be a technical argument for where the unpredictability in the source comes from and why there is confidence in the entropy source delivering sufficient entropy for the uses made of the RBG output (by this particular OS). This argument will include a description of the expected min-entropy rate (i.e. the minimum entropy (in bits) per bit or byte of source data) and explain that sufficient entropy is going into the OS randomizer seeding process. This discussion will be part of a justification for why the entropy source can be relied upon to produce bits with entropy.

The amount of information necessary to justify the expected min-entropy rate depends on the type of entropy source included in the product.

For developer provided entropy sources, in order to justify the min-entropy rate, it is expected that a large number of raw source bits will be collected, statistical tests will be performed, and the min-entropy rate determined from the statistical tests. While no particular statistical tests are required at this time, it is expected that some testing is necessary in order to determine the amount of min-entropy in each output.

For third-party provided entropy sources, in which the OS vendor has limited access to the design and raw entropy data of the source, the documentation will indicate an estimate of the amount of min-entropy obtained from this third-party source. It is acceptable for the vendor to "assume" an amount of min-entropy, however, this assumption must be clearly stated in the documentation provided. In particular, the min-entropy estimate must be specified and the assumption included in the ST.

Regardless of type of entropy source, the justification will also include how the DRBG is initialized with the entropy stated in the ST, for example by verifying that the min-entropy rate is multiplied by the amount of source data used to seed thDRBG or that the rate of entropy expected based on the amount of source data is explicitly stated and compared to the statistical rate. If the amount of source data used to seed the DRBG is not clear or the calculated rate is not explicitly related to the seed, the documentation will not be considered complete.

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

## D.3 Operating Conditions

The entropy rate may be affected by conditions outside the control of the entropy source itself. For example, voltage, frequency, temperature, and elapsed time after power-on are just a few of the factors that may affect the operation of the entropy source. As such, documentation will also include the range of operating conditions under which the entropy source is expected to generate random data. It will clearly describe the measures that have been taken in the system design to ensure the entropy source continues to operate under those conditions. Similarly, documentation shall describe the conditions under which the entropy source is known to malfunction or become inconsistent. Methods used to detect failure or degradation of the source shall be included.

## D.4 Health Testing

More specifically, all entropy source health tests and their rationale will be documented. This includes a description of the health tests, the rate and conditions under which each health test is performed (e.g., at start, 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~~ E - References

| Identifier | Title |
|---|---|
| [CC] | Common Criteria for Information Technology Security Evaluation -<br>• Part 1: Introduction and General Model, CCMB-2017-04-001, Version 3.1, Revision 5, April 2017.<br>• Part 2: Security Functional Components, CCMB-2017-04-002, Version 3.1, Revision 5, April 2017.<br>• 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-2012-09-004, Version 3.1, Revision 4, September 2012. |
| [CESG] | CESG - End User Devices Security and Configuration Guidance |
| [CSA] | Computer Security Act of 1987, H.R. 145, June 11, 1987. |
| [OMB] | Reporting Incidents Involving Personally Identifiable Information and Incorporating the Cost for Security in Agency Information Technology Investments, OMB M-06-19, July 12, 2006. |

# Appendix F - Acronyms

| Acronym | Meaning |
|---|---|
| AES | Advanced Encryption Standard |
| API | Application Programming Interface |
| API | Application Programming Interface |
| ASLR | Address Space Layout Randomization |
| Base-PP | Base Protection Profile |
| CC | Common Criteria |
| CEM | Common Evaluation Methodology |
| CESG | Communications-Electronics Security Group |
| CMC | Certificate Management over CMS |
| CMS | Cryptographic Message Syntax |
| CN | Common Names |
| CRL | Certificate Revocation List |
| CSA | Computer Security Act |
| CSP | Critical Security Parameters |
| DAR | Data At Rest |
| DEP | Data Execution Prevention |
| DES | Data Encryption Standard |
| DHE | Diffie-Hellman Ephemeral |
| DNS | Domain Name System |
| DRBG | Deterministic Random Bit Generator |
| DSS | Digital Signature Standard |
| DSS | Digital Signature Standard |
| DT | Date/Time Vector |
| DTLS | Datagram Transport Layer Security |
| EAP | Extensible Authentication Protocol |
| ECDHE | Elliptic Curve Diffie-Hellman Ephemeral |
| ECDSA | Elliptic Curve Digital Signature Algorithm |
| EST | Enrollment over Secure Transport |
| FIPS | Federal Information Processing Standards |
| HMAC | Hash-based Message Authentication Code |
| HTTP | Hypertext Transfer Protocol |
| HTTPS | Hypertext Transfer Protocol Secure |
| IETF | Internet Engineering Task Force |
| IP | Internet Protocol |
| ISO | International Organization for Standardization |
| IT | Information Technology |
| ITSEF | Information Technology Security Evaluation Facility |
| NIAP | National Information Assurance Partnership |

| | |
|---|---|
| NIST | National Institute of Standards and Technology |
| OCSP | Online Certificate Status Protocol |
| OE | Operational Environment |
| OID | Object Identifier |
| OMB | Office of Management and Budget |
| OS | Operating System |
| PII | Personally Identifiable Information |
| PKI | Public Key Infrastructure |
| PP | Protection Profile |
| PP | Protection Profile |
| PP-Configuration | Protection Profile Configuration |
| PP-Module | Protection Profile Module |
| RBG | Random Bit Generator |
| RFC | Request for Comment |
| RNG | Random Number Generator |
| RNGVS | Random Number Generator Validation System |
| S/MIME | Secure/Multi-purpose Internet Mail Extensions |
| SAN | Subject Alternative Name |
| SAR | Security Assurance Requirement |
| SFR | Security Functional Requirement |
| SHA | Secure Hash Algorithm |
| SIP | Session Initiation Protocol |
| ST | Security Target |
| SWID | Software Identification |
| TLS | Transport Layer Security |
| TOE | Target of Evaluation |
| TSF | TOE Security Functionality |
| TSFI | TSF Interface |
| TSS | TOE Summary Specification |
| URI | Uniform Resource Identifier |
| URL | Uniform Resource Locator |
| USB | Universal Serial Bus |
| VM | Virtual Machine |
| XCCDF | eXtensible Configuration Checklist Description Format |
| XOR | Exclusive Or |
| app | Application |