US GOVERNMENT PROTECTION PROFILE FOR GENERAL-PURPOSE OPERATING SYSTEMS IN A NETWORKED ENVIRONMENT

Version 1.0

Information Assurance Directorate

30 August 2010

Foreword

This publication, "*U.S. Government Protection Profile for General-Purpose Operating Systems in a Networked Environment",* is issued by the Information Assurance Directorate as part of its program to promulgate security standards for information systems.

Common Criteria Version:

This Protection Profile (PP) was written using Version 3.1 Revision 2 of the Common Criteria (CC).

Table of Contents

List of Figures

List of Tables

1. Introduction

This section contains overview information necessary to allow a Protection Profile (PP) to be registered through a Protection Profile Registry. The PP identification provides the labeling and descriptive information necessary to identify, catalogue, register, and cross-reference a PP. The PP overview summarizes the profile in narrative form and provides sufficient information for a potential user to determine whether the PP is of interest. The overview can also be used as a stand-alone abstract for PP catalogues and registers. The "Conventions" section provides the notation, formatting, and conventions used in this protection profile. The "Glossary of Terms" section gives a basic definition of terms, which are specific to this PP. The "Document Organization" section briefly explains how this document is organized.

1.1 Identification

Title: US Government Protection Profile for General-Purpose Operating Systems in a Networked Environment

Keywords: operating system, COTS, commercial security, access control, discretionary access control, DAC, cryptography.

1.2 Overview

The "*U.S. Government Protection Profile for General-Purpose Operating Systems in a Networked Environment"* specifies security requirements for commercial-off-the-shelf (COTS) general-purpose operating systems in networked environments. This profile establishes the requirements necessary to achieve the security objectives of the Target of Evaluation (TOE) and its environment.

Conformant products support Identification and Authentication, Discretionary Access Control (DAC), and an audit capability and Cryptographic Services. These systems provide adequate security services, mechanisms, and assurances to process administrative, private, and sensitive/proprietary information. When an organization's most sensitive/proprietary information is to be sent over a publicly accessed network, the organization should apply additional protection at the network boundaries.

1.3 Conventions

The notation, formatting, and conventions used in this PP are consistent with version 3.1 of the Common Criteria for Information Technology Security Evaluation. Font style and clarifying information conventions were developed to aid the reader.

The CC permits four functional component operations: assignment, iteration, refinement, and selection to be performed on functional requirements. These operations are defined in Common Criteria, Part 1, paragraph 6.4.1.3.2 as:

assignment: allows the specification of an identified parameter;

refinement: allows the addition of details or the narrowing of requirements;

selection: allows the specification of one or more elements from a list; and

iteration: allows a component to be used more than once with varying operations.

Assignments or selections left to be specified by the developer in subsequent security target documentation are italicized and identified between brackets ("[ ]"). In addition, when an assignment or selection has been left to the discretion of the developer, the text "assignment:" or "selection:" is indicated within the brackets. Assignments or selection created by the PP author (for the developer to complete) are bold, italicized, and between brackets ("[ ]"). CC selections completed by the PP author are underlined and CC assignments completed by the PP author are bold.

Refinements are identified with "**Refinement:**" right after the short name. They permit the addition of extra detail when the component is used. The underlying notion of a refinement is that of narrowing. There are two types of narrowing possible: narrowing of implementation and narrowing of scope[1]. Additions to the CC text are specified in bold. Deletions of the CC text are identified in the "End Notes" with a bold number after the element ("**8**").

Iterations are identified with a number inside parentheses ("(#)"). These follow the short family name and allow components to be used more than once with varying operations.

Extended Requirements are allowed to create requirements should the Common Criteria not offer suitable requirements to meet the PP needs. The naming convention for extended requirements is the same as that used in the CC. To ensure these requirements are identified, the word "**Extended:**" appears before the component behavior name to alert the reader. Additionally, the ending "**_EXT**" is appended to the newly created short name and the component and the element names are bolded. However, most of the extended requirements are based on existing CC requirements.

Application Notes are used to provide the reader with additional requirement understanding or to clarify the author's intent. These are italicized and usually appear following the element needing clarification.

Table 1.1 provides examples of the conventions (explained in the above paragraphs) for the permitted operations.

[1] US interpretation #0362: Scope of Permitted Refinements

Table 1.1 Functional Requirements Operation Conventions

Convention	Purpose	Operation
Bold	The purpose of bolded text is used to alert the reader that additional text has been added to the CC. This could be an assignment that was completed by the PP author or a refinement to the CC statement. Examples: FAU_SAR.1.1 The TSF shall provide **authorized administrators** with the capability to read **all audit information** from the audit records. FTA_MCS.1.1 **Refinement:** The TSF shall restrict the maximum number of concurrent **interactive** sessions that belong to the same user.	(Completed) Assignment or Refinement
Italics	The purpose of italicized text is to inform the reader of an assignment or selection operation to be completed by the developer or ST author. It has been left as it appears in the CC requirement statement. Examples: FTA_SSL.1.1 The TSF shall lock an interactive session after [*assignment: a time interval of user inactivity*] by: a) Clearing or overwriting display devices, making the current contents unreadable. b) Disabling any activity of the user's data access/display devices other than unlocking the session. FDP_RIP.2.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the [*selection: allocation of the resource to, deallocation of the resource from*] all objects.	Assignment (to be completed by developer or ST author) or Selection (to be completed by developer or ST author)
<u>Underline</u>	The purpose of underlined text is to inform the reader that a choice was made from a list provided by the CC selection operation statement. Example: FAU_STG.1.2 The TSF shall be able to <u>prevent</u> modifications to the audit records.	Selection (completed by PP author)

Convention	Purpose	Operation
Bold & Italics	The purpose of bolded and italicized text is to inform the reader that the author has added new text to the requirement and that an additional vendor action needs to be taken. Example: FIA_UAU.1.1 **Refinement:** The TSF shall allow **read access to [*assignment: list of public objects*]** on behalf of the user to be performed before the user is authenticated. FCS_CKM.2.1 The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method **[*selection: Manual (Physical) Method, Automated (Electronic Method), Manual Method and Automated Method*]** that meets the …	Assignment (added by the PP author for the developer or ST author to complete) or Selection (added by the PP author for the developer or ST author to complete)
Parentheses (Iteration #)	The purpose of using parentheses and an iteration number is to inform the reader that the author has selected a new field of assignments or selections with the same requirement and that the requirement will be used multiple times. Iterations are performed at the component level. The component behavior name includes information specific to the iteration between parentheses. Example: 1.3.1.1 5.5.3.1 Management of TSF Data (for general TSF data) (FMT_MTD.1(1)) FMT_MTD.1.1(1) The TSF shall restrict the ability to **create**, query, modify, delete, and clear the **security-relevant TSF data except for audit records, user security attributes and authentication data** to **the authorized administrator**. 1.3.1.2 5.5.3.2 Management of TSF Data (for audit records) (FMT_MTD.1(2)) FMT_MTD.1.1(2) The TSF shall restrict the ability to query, delete, and clear the **audit records** to **authorized administrators**.	Iteration 1 (of component) Iteration 2 (of component)

Convention	Purpose	Operation
Extended: (_EXT)	The purpose of using **Extended:** before the family or component behavior name is to alert the reader and to explicitly identify a newly created component. To ensure these requirements are identified as Extended, the "**_EXT**" is appended to the newly created short name and the component and element names are bolded. Example: 1.3.1.3 5.5.7.1 Extended: Internal TSF Data Consistency (FPT_TRC_EXT.1) **FPT_TRC_EXT.1.1** The TSF shall ensure that TSF data is consistent between parts of the TOE by providing a mechanism to bring inconsistent TSF data into a consistent state in a timely manner.	Extended Requirement
Endnotes	The purpose of endnotes is to alert the reader that the author has deleted Common Criteria text. An endnote number is inserted at the end of the requirement, and the endnote is recorded on the last page of the section. The endnote statement first states that a deletion was performed and then provides the rationale. Following is the family behavior or requirement in its original and modified form. A strikethrough is used to identify deleted text and bold for added text. A text deletion rationale is provided. Examples: Text as shown: FAU_ARP.1.1 **Refinement:** Upon detection of a potential security violation, the TSF shall **generate a warning message to the authorized administrator that requires explicit acknowledgement by the administrator.**[18] Endnote statement: [18] A deletion of CC text was performed in FAU_ARP.1.1. Rationale: The word "take" was deleted for clarity and better flow of the requirement. Additionally the words, "upon detection of a potential security violation" were moved to the beginning of the requirement to make requirement clearer. FAU_ARP.1.1 **Refinement:** Upon detection of a potential security violation, the TSF shall ~~take~~ **generate a warning to the authorized administrator** ~~upon detection of a potential security violation~~ **that requires explicit acknowledgement by the administrator.**	Refinement

1.4 Glossary of Terms

This profile uses the terms described in this section to aid in the application of the requirements.

Access	Interaction between an entity and an object that results in the flow or modification of data.
Access control	Security service that controls the use of resources[2] and the disclosure and modification of data[3].
Accountability	Tracing each activity in an IT system to the entity responsible for the activity.
Administrator	An authorized user who has been specifically granted the authority to manage some portion or the entire TOE and thus whose actions may affect the TSP. Administrators may possess special privileges that provide capabilities to override portions of the TSP.
Assurance	A measure of confidence that the security features of an IT system are sufficient to enforce its' security policy.
Attack	An intentional act attempting to violate the security policy of an IT system.
Authentication	Security measure that verifies a claimed identity.
Authentication data	Information used to verify a claimed identity.
Authorization	Permission, granted by an entity authorized to do so, to perform functions and access data.
Authorized user	An authenticated user who may, in accordance with the TSP, perform an operation.
Availability	Timely[4], reliable access to IT resources.
Compromise	Violation of a security policy.
Confidentiality	A security policy pertaining to disclosure of data.
Critical cryptographic security parameters	Security-related information appearing in plaintext or otherwise unprotected form and whose disclosure or modification can compromise the security of a cryptographic module or the security of the information protected by the module.

[2] hardware and software

[3] stored or communicated

[4] according to a defined metric

Cryptographic boundary	An explicitly defined contiguous perimeter that establishes the physical bounds (for hardware) or logical bounds (for software) of a cryptographic module.
Cryptographic key (key)	A parameter used in conjunction with a cryptographic algorithm that determines: – the transformation of plaintext data into ciphertext data, – the transformation of ciphertext data into plaintext data, – a digital signature computed from data, – the verification of a digital signature computed from data, or a data authentication code computed from data.
Cryptographic module	The set of hardware, software, and/or firmware that implements approved security functions (including cryptographic algorithms and key generation) and is contained within the cryptographic boundary.
Cryptographic module security policy	A precise specification of the security rules under which a cryptographic module must operate.
Defense-in-depth	A security design strategy whereby layers of protection are utilized to establish an adequate security posture for an IT system.
Discretionary Access Control (DAC)	A means of restricting access to objects based on the identity of subjects and groups to which they belong. The controls are discretionary in the sense that a subject with a certain access permission is capable of passing that permission (perhaps indirectly) on to any other subject.
Enclave	A collection of entities under the control of a single authority and having a homogeneous security policy. They may be logical, or based on physical location and proximity.
Entity	A subject, object, user or external IT device.
General-Purpose Operating System	A general-purpose operating system is designed to meet a variety of goals, including protection between users and applications, fast response time for interactive applications, high throughput for server applications, and high overall resource utilization.
Identity	A means of uniquely identifying an authorized user of the TOE.

Named object	An object that exhibits all of the following characteristics: - The object may be used to transfer information between subjects of differing user identities within the TSF. - Subjects in the TOE must be able to request a specific instance of the object. - The name used to refer to a specific instance of the object must exist in a context that potentially allows subjects with different user identities to request the same instance of the object.
Object	An entity under the control of the TOE that contains or receives information and upon which subjects perform operations.
Operating environment	The total environment in which a TOE operates. It includes the physical facility and any physical, procedural, administrative and personnel controls.
Persistent storage	All types of data storage media that maintain data across system boots (e.g., hard disk, CD, DVD).
Public object	An object for which the TSF unconditionally permits all entities "read" access. Only the TSF or authorized administrators may create, delete, or modify the public objects.
Resource	A fundamental element in an IT system (e.g., processing time, disk space, and memory) that may be used to create the abstractions of subjects and objects.
Secure State	Condition in which all TOE security policies are enforced.
Security attributes	TSF data associated with subjects, objects and users that is used for the enforcement of the TSP.
Security-enforcing	A term used to indicate that the entity (e.g., module, interface, subsystem) is related to the enforcement of the TOE security policies.
Security-supporting	A term used to indicate that the entity (e.g., module, interface, subsystem) is not security-enforcing however, its implementation must still preserve the security of the TSF.
Security Target (ST)	A set of security requirements and specifications to be used as the basis for evaluation of an identified TOE.
Subject	An active entity within the TSC that causes operations to be performed. Subjects can come in two forms: trusted and untrusted. Trusted subjects are exempt from part or all of the TOE security policies. Untrusted subjects are bound by all TOE security policies.
Target of Evaluation (TOE)	An IT product or system and its associated administrator and user guidance documentation that is the subject of an evaluation.

Threat	Capabilities, intentions and attack methods of adversaries, or any circumstance or event, with the potential to violate the TOE security policy.
User	Any person who interacts with the TOE.
Vulnerability	A weakness that can be exploited to violate the TOE security policy.

1.5 Document Organization

Section 1 provides the introductory material for the protection profile.

Section 2 describes the Target of Evaluation in terms of its envisaged usage and connectivity.

Section 3 defines the expected TOE security environment in terms of the threats to its security, the security assumptions made about its use, and the security policies that must be followed.

Section 4 identifies the security objectives derived from the threats and policies.

Section 5 identifies and defines the security functional requirements from the CC that must be met by the TOE in order for the functionality-based objectives to be met.

Section 6 identifies the security assurance requirements.

Section 7 provides a rationale to explicitly demonstrate that the information technology security objectives satisfy the policies and threats. Arguments are provided for the coverage of each policy and threat. The section then explains how the set of requirements are complete relative to the objectives, and that each security objective is addressed by one or more component requirements. Arguments are provided for the coverage of each objective.

Section 8 identifies background material used as reference to create this profile.

Appendix A defines frequently used acronyms.

Appendix B lists cryptographic standards, policies, and other related publications that have been identified in section 5.2 of this PP.

2. Target of Evaluation (TOE) Description

2.1 Product Type

This protection profile specifies requirements for general-purpose, multi-user, COTS operating systems for use in National Security Systems. Such operating systems are typically employed in a networked office automation environment (see Figure 2.1) containing file systems, printing services, network services and data archival services and can host other applications (e.g., mail, databases). This profile does not specify any security characteristics of security-hardened devices (e.g. guards, firewalls) that provide environment protection at network boundaries. **When this TOE is used in composition with other products to make up a larger system, the boundary protection must provide the appropriate security mechanisms.**

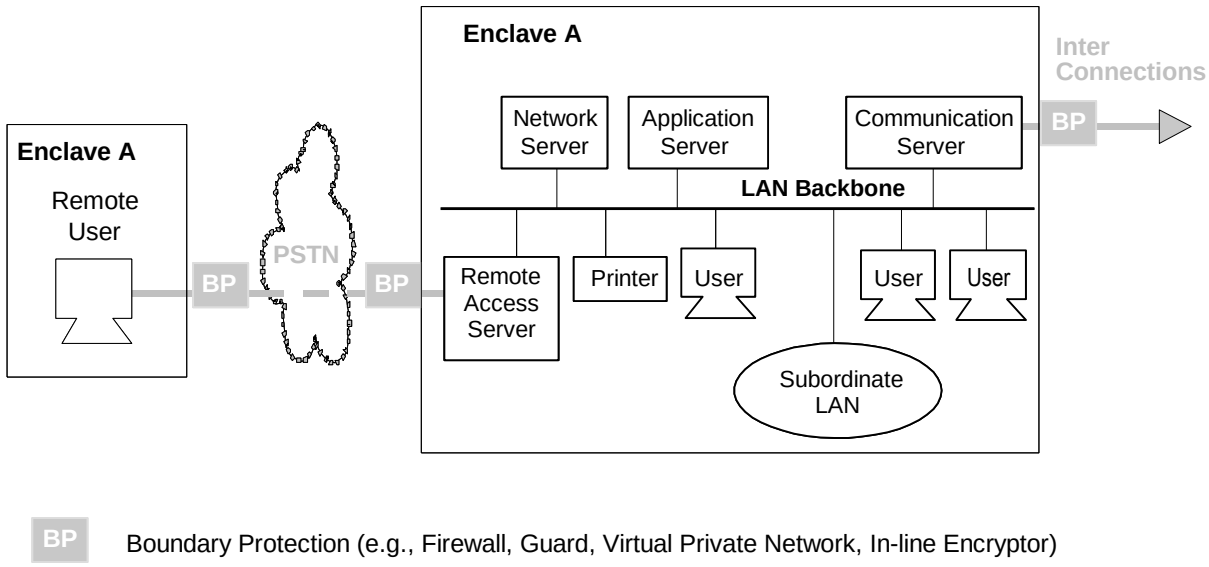

Figure 2-1 TOE Environment

2.2 Conformance Claims

1 This PP does not claim conformance to any other PPs. ST are able to claim demonstrable or strict conformance to this PP.

2.3 General TOE Functionality

Conformant operating systems include the following security features:

Identification and Authentication which mandates authorized users to be uniquely identified and authenticated before accessing information stored on the system;

Discretionary Access Control (DAC) which restricts access to objects based on the identity of subjects and groups to which they belong, and allows authorized users to specify protection for objects that they control;

Cryptographic services which provide mechanisms to protect TSF code and data and also provide support to allow authorized users and applications to encrypt, decrypt, hash, and digitally sign data as it resides within the system and as it is transmitted to other systems; and

Audit services which allow authorized administrators to detect and analyze potential security violations.

Requirements not addressed in this PP include:

- mechanisms or services to ensure availability of data residing on the TOE.[5],
- mechanisms or services to ensure integrity of user data residing on the TOE, and
- complete physical protection mechanisms, which must be provided by the environment.

2.4 Cryptographic Requirements

The TOE cryptographic services must provide both a level of functionality and assurance regardless of its implementation (software, hardware, or any combination thereof). This is achieved by meeting both the NIST FIPS PUB 140-2 standard and all additional requirements as stated in this PP (refer to Appendix B for relevant cryptographic standards, policies, and other publications).

2.5 TOE Operational Environment

It is assumed that the TOE environment is under the control of a single administrative authority and has a homogeneous system security policy, including personnel and physical security. This environment can be specific to an organization or a mission and may also contain multiple networks or enclaves. Enclaves may be logical or be based on physical location and proximity.

The TOE may be accessible by external IT systems that are beyond the environment's security policies. The users of these external IT systems are similarly beyond the control of the operating system's policies. Although the users of these external systems are authorized in their environments, they are outside the scope of control of this particular environment so nothing can be presumed about their intent. They must be viewed as potentially hostile.

This PP is appropriate for protection of administrative, private, and sensitive/proprietary information. When an organization's most sensitive information is to be sent over a publicly accessible network, the organization should consider applying additional layered security mechanisms.

[5] If availability requirements exist, the environment must provide the required mechanisms (e.g., mirrored/duplicated data).

3. TOE Security Environment

This section defines the expected TOE security environment in terms of the threats, security assumptions, and the security policies that must be followed for the TOE.

3.1 Threats

The following threats are addressed by PP compliant TOEs:

T.ADMIN_ERROR	An administrator may incorrectly install or configure the TOE resulting in ineffective security mechanisms.
T.ADMIN_ROGUE	An authorized administrator's intentions may become malicious resulting in user or TSF data being compromised.
T.AUDIT_COMPROMISE	A malicious user or process may view audit records, cause audit records to be lost or modified, or prevent future audit records from being recorded, thus masking a user's action.
T.CRYPTO_COMPROMISE	A malicious user or process may cause key, data or executable code associated with the cryptographic functionality to be inappropriately accessed (viewed, modified, or deleted), thus compromising the cryptographic mechanisms and the data protected by those mechanisms.
T.MASQUERADE	A malicious user, process, or external IT entity may masquerade as an authorized entity in order to gain unauthorized access to data or TOE resources.
T.OPERATIONAL_ERRORS	While the TOE is operational, changes to the TOE may cause it to enter a configuration that is not able to enforce the security policies of the TOE.
T.RESIDUAL_DATA	A user or process may gain unauthorized access to data through reallocation of TOE resources from one user or process to another.
T.RESOURCE_EXHAUSTION	A malicious process or user may block others from system resources (i.e., persistent storage) via a resource exhaustion denial of service attack.
T.TSF_COMPROMISE	A malicious user or process may cause TSF data or executable code to be inappropriately accessed (viewed, modified or deleted).
T.UNATTENDED_SESSION	A user may gain unauthorized access to an unattended session.

T.UNAUTHORIZED_ACCESS	A user may gain unauthorized access (view, modify, delete) to user data.
T.UNIDENTIFIED_ACTIONS	The administrator may fail to notice potential security violations, thus preventing the administrator from taking action against a possible security violation.
T.UNKNOWN_STATE	When the TOE is initially started or restarted after a failure, the security state of the TOE may be unknown.

3.2 Security Policy

The following organizational security policies are addressed by PP compliant TOEs:

P.ACCESS_BANNER	The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the TOE.
P.ACCOUNTABILITY	The users of the TOE shall be held accountable for their actions within the TOE.
P.AUTHORIZATION	The TOE shall limit the extent of each user's abilities in accordance with the TSP.
P.AUTHORIZED_USERS	Only those users who have been authorized to access the information within the TOE may access the TOE.
P.CRYPTOGRAPHY	The TOE shall use NIST FIPS validated cryptography as a baseline for key management (i.e., generation and destruction) and for cryptographic operations (i.e., encryption, decryption, signature, hashing, and random number generation services).
P.I_AND_A	All users must be identified and authenticated prior to accessing any controlled resources with the exception of public objects.
P.NEED_TO_KNOW	The TOE must limit the access to data in protected resources to those authorized users who have a need to know that data.
P.ROLES	The TOE shall provide multiple administrative roles for secure administration of the TOE. These roles shall be separate and distinct from each other.
P.TRACE	The TOE shall provide the ability to review the actions of individual users.

P.TRUSTED_RECOVERY	Procedures and/or mechanisms shall be provided to assure that, after a TOE failure or other discontinuity, recovery without a protection compromise is obtained.

3.3 Security Usage Assumptions

The specific conditions below are assumed to exist in a PP-compliant TOE environment:

A.PHYSICAL	It is assumed that the IT environment provides the TOE with appropriate physical security, commensurate with the value of the IT assets protected by the TOE.

4. Security Objectives

This section defines the security objectives for the TOE and its environment. These objectives are suitable to counter all identified threats and cover all identified organizational security policies and assumptions. The TOE security objectives are identified with "O." appended to the beginning of the name and the environment objectives are identified with "OE." appended to the beginning of the name.

4.1 TOE Security Objectives

O.ACCESS	The TOE will ensure that users gain only authorized access to it and to resources that it controls.
O.ACCESS_HISTORY	The TOE will display information (to authorized users) related to previous attempts to establish an interactive session.
O.ADMIN_ROLE	The TOE will provide administrator roles to isolate administrative actions.
O.AUDIT_GENERATION	The TOE will provide the capability to detect security relevant events and create records of those events in the audit trail.
O.AUDIT_PROTECTION	The TOE will provide the capability to protect audit information.
O.AUDIT_REVIEW	The TOE will provide the capability to selectively view audit information and alert the administrator of identified potential security violations.
O.CORRECT_TSF_OPERATION	The TOE will provide a capability to test the TSF to ensure the correct operation of the TSF in its operational environment.
O.CRYPTOGRAPHIC_SERVICES	The TOE will make encryption services available to authorized users and/or user applications.
O.DISCRETIONARY_ACCESS	The TOE will control access to named objects based upon the identity of users and groups of users.
O.DISCRETIONARY_USER_CONTROL	The TOE will allow authorized users to specify the named objects may be accessed by which users and groups of users.
O.DISPLAY_BANNER	The TOE will display (where appropriate) an advisory warning regarding use of the TOE.

O.MANAGE	The TOE will provide all the functions and facilities necessary to support the authorized administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.
O.PROTECT	The TOE will provide mechanisms to protect user data and resources.
O.RECOVERY	Procedures and/or mechanisms will be provided to assure that recovery is obtained without a protection compromise, such as from system failure or discontinuity.
O.RESIDUAL_INFORMATION	The TOE will ensure that any data contained in a protected resource is not available when the resource is reallocated.
O.RESOURCE_EXHAUSTION	The TOE shall provide mechanisms that mitigate user attempts to exhaust persistent storage.
O.DOMAIN_ISOLATION	The TOE will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure.
O.TSF_CRYPTOGRAPHIC_INTEGRITY	The TOE will provide cryptographic integrity mechanisms for TSF data while in transit to remote parts of the TOE.
O.USER_AUTHENTICATION	The TOE will verify the claimed identity of users.
O.USER_IDENTIFICATION	The TOE will uniquely identify users.

4.2 Environment Security Objectives

OE.PHYSICAL	Physical security will be provided for the TOE by the IT environment, commensurate with the value of the IT assets protected by the TOE.

5. Security Functional Requirements

This section contains detailed security functional requirements for the operating systems' trusted security functions (TSF) of general-purpose COTS operating systems. The requirements contained in this section are either selected from Part 2 of the CC or are extended components (with short names ending in "**_EXT**"). Table 5.1 lists the extended functional requirements in this section.

Table 5.1 Extended Functional Requirements

Extended Component	Component Behavior Name
FCS_BCM_EXT.1	Baseline Cryptographic Module
FCS_COA_EXT.1	Cryptographic Operations Availability
FCS_RGB_EXT.1	Random Number Generation
FPT_TRC_EXT.1	Internal TSF Data Consistency
FPT_TST_EXT.1	TSF Testing

5.1 Security Audit (FAU)

5.1.1 Security Audit Data Generation (FAU_GEN)

5.1.1.1 Audit Data Generation (FAU_GEN.1)

Dependencies: FPT_STM.1 Reliable time stamps

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 the audit functions,

b) **Start-up and shutdown of the TOE,**

c) **Uses of special permissions that circumvent the access control policies,**

Application Note: These special permissions are typically those often used by authorized administrators.

d) **All auditable events listed in Table 5.2, and**

e) All auditable events for the minimal level of audit.

Application Note: For other security relevant functions that are not included in this PP, the ST author defines a minimal level of audit.

Table 5.2 Auditable Events

Requirement	Audit events prompted by requirement	Additional Information in audit record (FAU_GEN.1.2b)
Audit Data Generation (FAU_GEN.1)	(none)	(none)

User Identity Association (FAU_GEN.2)	(none)	(none)
Audit Review (FAU_SAR.1)	• Opening the audit records.	Name of object (audit log file)
Restricted Audit Review (FAU_SAR.2)	• Unsuccessful attempts to read information from the audit records.	(none)
Selectable Audit Review (FAU_SAR.3)	(none)	(none)
Selective Audit (FAU_SEL.1)	• All modifications to the audit configuration that occur while the audit collection functions are operating.	(none)
Protected Audit Trail Storage (FAU_STG.1)	(none)	(none)
Action in case of possible audit data loss (FAU_STG.3)	• Actions taken due to exceeding of a threshold.	Message sent to administrator
Extended: Baseline Cryptographic Module (FCS_BCM_EXT.1)	• Failure of the cryptographic operation.	(none)
Cryptographic Key Generation (FCS_CKM.1)	• Failure of the key generation process.	(none)
Cryptographic Key Destruction (FCS_CKM.4)	• Failure of key zeroization process.	Identity of subject requesting or causing zeroization, identity of object or entity being cleared.
Extended: Cryptographic Operations Availability (FCS_COA_EXT.1)	(none)	(none).
Cryptographic Operation (for data encryption/decryption) (FCS_COP.1(1))	• Failure in encryption or decryption.	Cryptographic mode of operation, name of object being encrypted/decrypted.
Cryptographic Operation (for cryptographic signature) (FCS_COP.1(2))	• Failure in cryptographic signature.	Cryptographic mode of operation, name of object being signed/verified.
Cryptographic Operation (for cryptographic hashing) (FCS_COP.1(3))	• Failure in hashing function.	Cryptographic mode of operation, name of object being hashed.
Extended: Random Number Generation (FCS_RBG_EXT.1)	• Failure in the randomization process.	(none)
Subset Access Control (FDP_ACC.1)	(none)	(none)
Security Attribute Based Access Control (FDP_ACF.1)	• All requests to perform an operation on an object covered by the SFP. • Use of privilege to bypass the access control mechanism.	The name of the object being accessed.

Subset Residual Information Protection (FDP_RIP.1)	(none)	(none)
Authentication Failure Handling (FIA_AFL.1)	• The reaching of the threshold for the unsuccessful authentication attempts • the action taken (disable for non-administrators, delay for administrator) • the re-enablement of disabled non-administrative accounts	(none)
User Attribute Definition (FIA_ATD.1)	(none)	(none)
Verification of Secrets (FIA_SOS.1)	(none)	(none)
Timing of Authentication (FIA_UAU.1)	• All use of the authentication mechanism.	Origin of the attempt (e.g., terminal identifier, source IP address)
Re-authenticating (FIA_UAU.6)	• All re-authentication attempts when changing authentication data	Origin of the attempt (e.g., terminal identifier, source IP address)
Protected Authentication Feedback (FIA_UAU.7)	(none)	(none)
Timing of Identification (FIA_UID.1)	• All use of the user identification mechanism	Provided user identity, origin of the attempt (e.g., terminal identifier, source IP address)
User-Subject Binding (FIA_USB.1)	• Binding of user security attributes to a subject (e.g. creation of a subject).	(none)
Management of Security Functions Behavior (for specification of auditable events) (FMT_MOF.1(1))	• All modifications in the behavior of the functions in the TSF.	The old and new values for audit events specified by this function.
Management of Security Functions Behavior (for authentication data) (FMT_MOF.1(2))	• All modifications in the behavior of the functions in the TSF.	(none)
Management of Security Attributes (FMT_MSA.1)	• All modifications of the values of security attributes.	The name of the object, the old and new values of the attributes
Secure Security Attributes (FMT_MSA.2)	• All modifications of the values of security attributes.	• All offered and rejected values for a security attribute.
Static Attributes Initialization (FMT_MSA.3)	• Modifications of the default setting of permissive or restrictive rules. • All modifications of the initial values of security attributes.	The old and new values of the attributes.
Management of TSF Data (for general TSF data) (FMT_MTD.1(1))	• All modifications of the values of TSF data.	The old and new values of the TSF data.

Management of TSF Data (for audit data) (FMT_MTD.1(2))	• Actions taken with respect to the audit records	The specific action that was performed.
Management of TSF Data (for initialization of user security attributes) (FMT_MTD.1(3))	• All initializations of the values of user security attributes.	The initial values for the user security attributes.
Management of TSF Data (for modification of user security attributes, other than authentication data) (FMT_MTD.1(4))	• All modifications of the values of user security attributes.	The old and new values of the attributes.
Management of TSF Data (for modification of authentication data) (FMT_MTD.1(5))	• All actions associated with modifications of the values of authentication data.	(none)
Management of TSF Data (for reading of authentication data) (FMT_MTD.1(6))	(none)	(none)
Management of TSF Data (for critical cryptographic security parameters) (FMT_MTD.1(7))	• All actions associated with modifications of the values of critical cryptographic security parameters.	The old and new values of the parameters, excluding any sensitive information, such as secret or private keys.
Revocation (to authorized administrators) (FMT_REV.1(1))	• All attempts to revoke security attributes.	The security attributes that are attempting to be revoked
Revocation (to owners and authorized administrators) (FMT_REV.1(2))	• All attempts to revoke security attributes.	The security attributes that are attempting to be revoked, the object with which the security attributes are associated.
Time-Limited Authorization (FMT_SAE.1)	• Specification of the expiration time for an attribute. • Action taken due to attribute expiration.	(none)
Specification of Management Functions (FMT_SMF.1)	(none)	(none)
Security Roles (FMT_SMR.1)	• Modifications to the group of users that are part of a role.	The role the user is associated with or disassociated from.
Basic Internal TSF Data Transfer Protection (FPT_ITT.1)	(none)	(none)
TSF Data Integrity Monitoring (FPT_ITT.3)	• Detection of modification of TSF data.	Network address of source and destination of the transfer.
Manual Recovery (FPT_RCV.1)	• The fact that a failure or service discontinuity occurred. • Resumption of the regular operation.	• Type of failure or service discontinuity
Reliable Time Stamps (FPT_STM.1)	• Setting the time to a specific value.	The old and new values for the time.

Extended: Internal TSF Data Consistency (FPT_TRC_EXT.1)	(none)	(none)
TSF Testing (for cryptography) (FPT_TST.1)	• Execution of the cryptography self tests.	For each test, the identification of the test and the results of that test.
Maximum Quotas (FRU_RSA.1)	• Rejection of allocation operation due to persistent storage limits.	Object or other entity associated with failed allocation operation.
Basic limitation on multiple concurrent sessions (FTA_MCS.1)	• Rejection of a new session based on the limitation of multiple concurrent sessions. • Setting the limit on the number of multiple concurrent sessions by an authorized administrator.	The old and new values of the number of multiple concurrent sessions (for setting the session limit).
TSF-Initiated Session Locking (FTA_SSL.1)	• Locking of an interactive session by the session locking mechanism. • Any attempts at unlocking of an interactive session.	(none)
User-Initiated Locking (FTA_SSL.2)	• Locking of an interactive session by the session locking mechanism. • Any attempts at unlocking of an interactive session.	(none)
Default TOE Access Banners (FTA_TAB.1)	(none)	(none)
TOE Access History (FTA_TAH.1)	(none)	(none)

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

a) Date and time of the event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event; and

Application Note: "Subject identity" means user identity associated with the subject.

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, **the additional information in Table 5.2.**

Application Note: Other audit relevant information associated with security-relevant functions not included in this PP should be included within the audit records.

5.1.1.2 User Identity Association (FAU_GEN.2)

Dependencies: FAU_GEN.1 Audit data generation
FIA_UID.1 Timing of identification

FAU_GEN.2.1 For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

Application Note: For failed login attempts no user identity association is required because the user is not under TSF control until after a successful identification/authentication.

5.1.2 Security Audit Review (FAU_SAR)

5.1.2.1 Audit Review (FAU_SAR.1)

Dependencies: FAU_GEN.1 Audit data generation

FAU_SAR.1.1 The TSF shall provide **authorized administrators** with the capability to read **all audit information** from the audit records.

Application Note: For a distributed system, the authorized administrator should be able to read all audit information within the TOE.

FAU_SAR.1.2 **Refinement:** The TSF shall provide the audit records in a manner suitable for the **authorized administrator** to interpret the information **using a tool to access the audit records.1**

Application Note: The tool provides a means to easily and efficiently review the audit records. It is expected (yet not necessary) that the tool satisfying this requirement will also satisfy the FAU_SAR.3 requirement.

5.1.2.2 Restricted Audit Review (FAU_SAR.2)

Dependencies: FAU_SAR.1 Audit review

FAU_SAR.2.1 The TSF shall prohibit all users read access to the audit records, except those users that have been granted explicit read-access.

5.1.2.3 Selectable Audit Review (FAU_SAR.3)

Dependencies: FAU_SAR.1 Audit review

FAU_SAR.3.1 **Refinement:** The TSF shall provide the ability to **perform searches** of audit data based on **the following attributes: 2**

a) user identity,
b) object identity,
c) date of the event,
d) time of the event,
e) type of event,
f) success of auditable security events, and
***g)* failure of auditable security events.**

5.1.3 Security Audit Event Selection (FAU_SEL)

5.1.3.1 Selective Audit (FAU_SEL.1)

Dependencies: FAU_GEN.1 Audit data generation
FMT_MTD.1 Management of TSF data

FAU_SEL.1.1 **Refinement:** The TSF shall be able to **include or exclude auditable** events from the set of **audited** events based on the following attributes:**3**

a) object identity,
b) user identity,
c) host identity,

d) event type,

e) **success of auditable security events, and**

f) **failure of auditable security events.**

Application Note: Each item listed in Table 5.2, 2nd column is an event and is searchable with respect to this requirement. However, multiple events can be combined into one audit record (for instance, use of the user identification and authentication mechanisms, while two events in table 5.2, will likely be combined into a single audit record).

5.1.4 Security Audit Event Storage (FAU_STG)

5.1.4.1 Protected Audit Trail Storage (FAU_STG.1)

Dependencies: FAU_GEN.1 Audit data generation

FAU_STG.1.1 The TSF shall protect the stored audit records in the audit trail from unauthorized deletion.

FAU_STG.1.2 **Refinement:** The TSF shall be able to prevent modifications to the stored audit records in the audit trail.**4**

Application Note: In order to reduce the performance impact of audit generation, audit records are often temporarily buffered in memory before being written to the disk. In such implementations, these buffered records will be lost if the operation of the TOE is interrupted by hardware or power failures. The developer should document the expected loss in such circumstances and show that it has been minimized.

5.1.4.2 Action in case of possible audit data loss (FAU_STG.3)

Dependencies: FAU_STG.1 Protected audit trail storage

FAU_STG.3.1 The TSF shall **notify an authorized administrator of the possible audit data loss** if the audit trail exceeds **an authorized administrator selectable, pre-defined limit**.

5.2 Cryptographic Support (FCS)

This section specifies the cryptographic support required in the TOE. Evolving public standards on cryptographic functions and related areas have required an interim approach to writing cryptographic requirements for general purpose operating systems. These cryptographic requirements are expected to be achievable in commercial products in the near term, and gradually mature over time. Today these requirements represent a step in the direction of helping to improve the security in COTS products. Over time, the Protection Profile will be updated as the underlying public standards and the body of related special publications mature.

5.2.1 Baseline Cryptographic Module (FCS_BCM)

The cryptographic requirements are structured to accommodate use of the FIPS 140-2 standard and NIST's Cryptomodule Validation Program (CMVP) in meeting the requirements. Note that *FIPS-approved* cryptographic functions are required to be implemented in a *FIPS-validated module running in FIPS-approved mode*. FCS_BCM reflects this requirement, and it specifies the required FIPS validation levels for the security functions. Note also that some of the requirements of this PP go beyond what is required for FIPS 140-2 validation. In these cases, Assurance Activities indicate any analysis/testing that is required to be performed by the CCTL.

The term "FIPS-approved cryptographic function" is used in the following requirements. A FIPS-approved cryptographic function is a security function (e.g., cryptographic algorithm, cryptographic key management technique, or authentication technique) that is either: 1) specified in a Federal Information Processing Standard (FIPS), or 2) adopted in a FIPS and specified either in an appendix to the FIPS or in a document referenced by the FIPS.

5.2.1.1 Extended: Baseline Cryptographic Module (FCS_BCM_EXT.1)

Dependencies: No dependencies.

FCS_BCM_EXT.1.1 All FIPS-approved cryptographic functions implemented by the TSF shall be implemented in a cryptomodule that is FIPS 140-2 validated, and perform the specified cryptographic functions in a FIPS-approved mode of operation. The FIPS 140-2 validation shall include an algorithm validation certificate for all FIPS-approved cryptographic functions.

Application Note: This Protection Profile shall use the term "FIPS 140-2" for simplicity. FIPS PUB 140-2 is currently undergoing a regular five year review; in the near future, FIPS PUB 140-3 will supersede it. Security Targets written to comply with this Protection Profile may replace it with the successor standard that is in force at the time of evaluation.

Application Note: This requirement does not preclude additional cryptographic algorithms from being implemented in the cryptomodule, and/or used by the TOE for purposes OTHER than those explicitly stated in this Protection Profile.

5.2.2 Cryptographic Key Management (FCS_CKM)

NIST Special Publication 800-57, "Recommendation for Key Management" contains additional protection mechanisms that vendors are encouraged to implement. It should also be used as guidance for the cryptographic key management requirements.

5.2.2.1 Cryptographic Key Generation (for symmetric keys) (FCS_CKM.1(1))

Dependencies: [FCS_RBG_EXT.1 Random number generation]
FCS_CKM.4 Cryptographic key destruction

FCS_CKM.1.1(1) **Refinement:** The TSF shall generate cryptographic keys **using a FIPS-Approved Random Number Generator as specified in FCS_RBG_EXT.1, and provide integrity protection to generated keys that leave the cryptomodule in accordance with NIST SP 800-57 "Recommendation for Key Management—Part 1: General," paragraph 6.2.2.2a. in the following manner: [assignment: *cryptographic integrity mechanism*].**

Application Note: For the assignment, the ST author includes the cryptographic integrity mechanism that is used to provide integrity protection on keys that leave the cryptomodule. Keys that do not leave the cryptomodule are assumed to be protected by the cryptomodule Examples of appropriate mechanisms are provided in 800-57. If the mechanism used is not already specified in the ST, the appropriate FCS requirements must be added to the ST to specify the integrity mechanisms.

5.2.2.2 Cryptographic Key Generation (for asymmetric keys) (FCS_CKM.1(2))

Dependencies: [FCS_RBG_EXT.1 Random number generation]
FCS_CKM.4 Cryptographic key destruction

FCS_CKM.1.1(2) **Refinement** The TSF shall generate **asymmetric** cryptographic keys in accordance with **domain parameter sizes [selection: for rDSA-based keys, [assignment:** *1024 bits or greater***], for ECDSA-based keys, [selection: 256 bits, 384 bits, 512 bits] t**hat meet the following: **FIPS 140-2.**

Application Note: This component requires that the TOE be able to generate the public/private key pairs that are used for the digital signature operations in FCS_COP.1(2). Therefore, the ST author must ensure that the selections and assignments correspond to that requirement in the ST. If multiple schemes are supported, then the ST author should iterate this requirement and FCS_COP.1(2) to capture this capability.

This requirement (along with FCS_BCM_EXT) also specifies that the key generation must be performed by a FIPS-validated cryptomodule operating in FIPS mode. For previously-validated cryptomodules, this can be verified by examination of the CMVP and CAVP validation lists maintained by NIST. The CMVP list will reference the digital signature algorithm used and a CAVP certificate number. The CAVP lists specify the algorithm implementations that perform key generation in addition to the signature operations for each signature scheme (RSA, ECDSA); the supported key generation claims must conform to the selections and assignments in this requirement in order for the TOE to claim compliance to this PP.

For the first selection, the ST author chooses the algorithm corresponding to the selection in FCS_COP.1(2); as noted above, if multiple algorithms are supported, this requirement should be iterated and one selection performed for each requirement. However, if multiple key sizes are implemented for a single algorithm (e.g., the implementation supports both P-256 and P-512 for ECDSA) then these should all be included in the same iteration. For the rDSA keys, the ST author fills in the assignment for the key sizes that are supported by the validated cryptomodule; these must be at least 1024 bits. For the ECDSA keys, the ST author selects one or more of the key sizes corresponding to the curves specified in FCS_COP.1(2).

5.2.2.3 Cryptographic Key Destruction (FCS_CKM.4)

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]

FCS_CKM.4.1 **Refinement:** The TSF shall destroy cryptographic keys in accordance with a **cryptographic key zeroization method** that meets the following: Key zeroization requirements of FIPS PUB 140-2, “Security Requirements for Cryptographic Modules”.

5.2.3 Cryptographic Operations Availability (FCS_COA)

5.2.3.1 Extended: Cryptographic Operations Availability (FCS_COA_EXT.1)

Dependencies: FCS_BCM_EXT.1 Extended: Baseline cryptographic module

FCS_COA_EXT.1 The TSF shall provide the following cryptographic operations to applications:

a) Encryption/Decryption,
b) Cryptographic Signature (Digital Signature),
c) Hashing, and

d) [*assignment: any other cryptographic operations provided to applications*].

Application Note: Combinations of these operations are also permissible. For instance, an encryption mode such as Galois Counter Mode which provides both encryption and data integrity (which is normally provided via secure hashing), is allowed.

5.2.4 Cryptographic Operation (FCS_COP)

5.2.4.1 Cryptographic Operation (for data encryption/decryption) (FCS_COP.1(1))

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1(1) **Refinement:** The TSF shall perform **encryption and decryption using the FIPS-approved security function AES algorithm operating in [*assignment: one or more FIPS-approved modes*]** and cryptographic key size **of [*selection: one or more of 128 bits, 192 bits, 256 bits*] that meets FIPS 140-2.**

5.2.4.2 Cryptographic Operation (for cryptographic signature) (FCS_COP.1(2))

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1(2) **Refinement:** The TSF shall perform **cryptographic signature services using the FIPS-approved security function [*selection:***

RSA Digital Signature Algorithm (rDSA) with a key size (modulus) of [assignment: 1024 bits or greater], or

Elliptic Curve Digital Signature Algorithm (ECDSA) with a key size of [selection: one or more of 256 bits, 384 bits, 521 bits], using only the NIST curve(s) [selection: one or more of P-256, P-384, P-521 as defined in FIPS PUB 186-3, "Digital Signature Standard"] **]**

that meets FIPS 140-2.

Application Note: For elliptic curve-based schemes, the key size refers to the $\log_2$ of the order of the base point. As the preferred approach for digital signatures, elliptic curves will be required after all the necessary standards and other supporting information are fully established.

5.2.4.3 Cryptographic Operation (for cryptographic hashing) (FCS_COP.1(3))

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1(3) **Refinement:** The TSF shall perform cryptographic hashing services in accordance with [***selection: SHA 256, SHA 384, SHA 512***] and message digest sizes [***selection: 256, 384, or 512***] bits that meet the following: **FIPS 140-2.**

Application Note: The message digest size should correspond to double the system symmetric

encryption key strength.

5.2.4.4 Extended: Random Number Generation (FCS_RBG_EXT.1)

Dependencies: FCS_BCM_EXT.1 Extended: Baseline cryptographic module

FCS_RBG_EXT.1.1 The TSF shall perform all random bit generation (RBG) services in accordance with [***selection: choose one of: NIST Special Publication 800-90, FIPS Pub 140-2 Annex C***] implemented in a FIPS-validated cryptomodule operating in FIPS mode seeded by an entropy source that accumulates entropy from [***selection: choose one of:***

one or more independent hardware-based noise sources,
one or more independent software-based noise sources,
a combination of hardware-based and software-based noise sources.*]*

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded with a minimum of **[*selection, choose one of: 128 bits, 192 bits, 256 bits*]** of entropy at least equal to the greatest bit length of the keys that it will generate.

5.3 User Data Protection (FDP)

5.3.1 Access Control Policy (FDP_ACC)

5.3.1.1 Subset Access Control (FDP_ACC.1)

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1 The TSF shall enforce the **Discretionary Access Control policy** on **all subjects and all named objects** and all operations among **them**.

Application Note: The DAC policy does not cover public objects.

5.3.2 Access Control Functions (FDP_ACF)

5.3.2.1 Security Attribute Based Access Control (FDP_ACF.1)

Dependencies: FDP_ACC.1 Subset access control
FMT_MSA.3 Static attribute initialization

FDP_ACF.1.1 **Refinement:** The TSF shall enforce the **Discretionary Access Control policy** to **named** objects based on the following **types of subject and object security attributes:**

a) the authorized user identity and group membership(s) associated with a subject;

b) the [authorized user (or group) identity, access operations] pairs associated with a named object; and

c) [selection: no other attributes, [assignment: other attributes used in access control decisions]]

Application Note: If no other attributes are used in making access control decisions, then "no other attributes" should be selected. Otherwise, the list of additional access control attributes should be included, and appropriate modifications made to the FMT_MSA requirements.

FDP_ACF.1.2 **Refinement:** The TSF shall enforce the following rules to determine if an operation among subjects and **named** objects is allowed:**5**

The Discretionary Access Control policy mechanism shall, either by explicit authorized user action or by default, provide that named objects are protected from unauthorized access according to the following ordered rules:

1) **If the requested mode of access is denied to that authorized user, deny access.**
2) **If the requested mode of access is permitted to that authorized user, permit access.**
3) **If the requested mode of access is denied to every group of which the authorized user is a member, deny access**
4) **If the requested mode of access is permitted to any group of which the authorized user is a member, grant access**
5) **If there is no rule explicitly allowing access, deny access.**

Application Note: This element specifies minimum granularity of access control functionality. It is not meant to preclude more fine grained access control mechanisms or additional rules inserted into the above set. However any more fine grained mechanisms must be capable of meeting the above rules, and any additional rules must result in access to named objects that is at least as restrictive as would be the case for the baseline set of rules above. For example, discretionary access rules on a file may be defined to take precedence over discretionary access rules on the directories containing that file.

FDP_ACF.1.3 **Refinement:** The TSF shall explicitly authorize access of subjects to **named** objects based on the following additional rules:

a) Authorized administrators must follow the above-stated Discretionary Access Control policy, except after taking the following specific actions: [*assignment: list of specific actions*],

b) The enforcement mechanism (e.g., access control lists) shall allow authorized users to specify and control sharing of named objects by individual user identities and group identities, and

c) [*assignment: other rules that explicitly authorize access of subjects to* **named** *objects*].

Application Note: This element allows specifications of additional rules for authorized administrators to bypass the Discretionary Access Control policy for system management or maintenance (e.g., system backup).

FDP_ACF.1.4 **Refinement:** The TSF shall explicitly deny access of subjects to **named** objects based on the **following rules**: [*assignment: rules that explicitly deny access of subjects to* **named** *objects*]**.**

5.3.3 Residual Information Protection (FDP_RIP)

5.3.3.1 Full Residual Information Protection (FDP_RIP.2)

Dependencies: No dependencies.

FDP_RIP.2.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon **[selection: allocation to, de-allocation from]** all objects.

Application Note: The ST author needs to consider all of the resources on the system, and document whether they are cleared on allocation or de-allocation. This will likely result in this

requirement being iterated in the ST.

5.4 Identification and Authentication (FIA)

5.4.1 Authentication Failures (FIA_AFL)

5.4.1.1 Authentication Failure Handling (FIA_AFL_EXT.1)

Dependencies: FIA_UAU.1 Timing of authentication

FIA_AFL_EXT.1.1 The TSF shall detect when an authorized administrator configurable positive integer of consecutive unsuccessful authentication attempts occur related to any authorized user authentication process.

FIA_AFL_EXT.1.2 When the defined number of consecutive unsuccessful authentication attempts has been met or surpassed, the TSF shall:

a) For all administrator accounts, "disable" the account for an authorized administrator configurable time period such that there can be no more than ten attempts per minute.

Application Note: Actually "disabling" the account is not required; the goal is to rate-limit the authorization attempts on an administrative account.

b) For all other accounts, disable the user logon account until it is re-enabled by the authorized administrator.

Application Note: The ability to disable user accounts is necessary to counter brute force discovery of the authentication data.

c) For all disabled accounts, any response to an authentication attempt given to the user shall not be based on the result of that authentication attempt.

Application Note: For item c above, the intent is that an attacker cannot get any information when they are attempting to brute force a password on a disabled account. For instance, if an attacker was returned the message "failed attempt" when they guessed an incorrect password on a disabled account, but was returned the message "account disabled" when a correct password was given on a disabled account, they have effectively guessed the password. This can be corrected in a number of ways. For instance, giving a "constant" message (e.g., "fail…fail…fail.[account disabled]…fail..fail…[correct password given] fail) or even not performing an authentication check on a disabled account would meet the requirement. It is also acceptable to change the message when the account becomes disabled, but the message must not reveal status of the authentication attempt (e.g., "fail…fail…fail…[account disabled]…locked out…locked out…[correct password given] locked out).

Application Note: "Consecutive unsuccessful authentication attempts" is the total number of unsuccessful attempts that occur, in order, prior to a successful authentication attempt. For distributed systems, this means that unsuccessful attempts from any node would contribute to the "consecutive failed attempts" count. However, FPT_TRC_EXT recognizes that there may be circumstances where the distributed nature of the TOE may cause a slight delay in accumulating these counts; this aspect should be documented in the TSS section and analyzed by the evaluators

to determine if it meets the intent of this requirement and the given assurance level of the TOE.

5.4.2 User Attribute Definition (FIA_ATD)

5.4.2.1 User Attribute Definition (FIA_ATD.1)

Dependencies: No dependencies.

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users:

- **a) unique identifier,**
- **b) group memberships,**
- **c) authentication data,**
- **d) security-relevant roles (see FMT_SMR.2),**
- **e) [*assignment: Any security attributes related to cryptographic function (e.g., certificate used to represent the user)*], and**
- **f) [*assignment: Any other security-relevant authorizations or attributes (e.g., privilege)*].**

Application Note: Group membership may be expressed in a number of ways: a list per user specifying to which groups the user belongs, a list per group which includes which users are members, or implicit association between certain user identities and certain groups.

Application Note: A TOE may have two forms of user and group identities which have a unique mapping between the representations.

Application Note: It is possible that the notion of privilege is tied to the security-relevant roles (item d).

5.4.3 Specification of Secrets (FIA_SOS)

5.4.3.1 Verification of Secrets (FIA_SOS.1)

Dependencies: No dependencies.

FIA_SOS.1.1 The TSF shall provide a mechanism to verify that secrets meet **the following:**

- **a) Passwords are at least 16 characters in length, consisting of any combination of upper and lower case letters, numbers, and symbols, and**
- **b) Passwords are not reused within the last administrator-settable number of passwords used by that user.**

Application Note: For item b, the TSF provides a mechanism for the administrator to set a password history such that a user cannot reuse any password that is on the password history list.

5.4.4 User Authentication (FIA_UAU)

5.4.4.1 Timing of Authentication (FIA_UAU.1)

Dependencies: FIA_UID.1 Timing of identification

FIA_UAU.1.1 The TSF shall allow **read access to public objects** on behalf of the user to be performed before the user is authenticated.

FIA_UAU.1.2 **Refinement:** The TSF shall require each user to be successfully authenticated **(i.e., an exact match between the internal representation of the user's entered data and the stored TSF authentication data)** before allowing any other TSF-mediated actions on behalf of that user.

Application Note: The entire entered user's authentication data must exactly match the entire stored data. No other parameters such as length of password should be used to short-circuit the authentication verification.

5.4.4.2 Re-authenticating (FIA_UAU.6)

Dependencies: No dependencies.

FIA_UAU.6.1 **Refinement:** The TSF shall re-authenticate the user **when changing authentication data.6**

Application Note: If the TOE is requiring the user to change authentication data upon having just authenticated (e.g., initial logon, session unlock), the user is considered to be re-authenticated.

5.4.4.3 Protected Authentication Feedback (FIA_UAU.7)

Dependencies: FIA_UAU.1 Timing of authentication

FIA_UAU.7.1 The TSF shall provide only **obscured feedback** to the user while the authentication is in progress.

Application Note: "Obscured feedback" implies the TSF does not produce a visible display of any authentication data entered by a user (such as the echoing of a password), although an obscured indication of progress may be provided (such as an asterisk for each character). It also implies that the TSF does not return any information during the authentication process to the user that may provide any indication of the authentication data.

5.4.5 User Identification (FIA_UID)

5.4.5.1 Timing of Identification (FIA_UID.1)

FIA_UID.1.1 The TSF shall allow **read access to *public objects*** on behalf of the user to be performed before the user is identified.

FIA_UID.1.2 The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

5.4.6 User-Subject Binding (FIA_USB)

5.4.6.1 User-Subject Binding (FIA_USB.1)

Dependencies: FIA_ATD.1 User attribute definition

FIA_USB.1.1 The TSF shall associate the following user security attributes with subjects acting on behalf of that user: **The security attributed identified in FIA_ATD.1a, b, d, and [assignment: *other attributes specified in assignments FIA_ATD.1e, f that subjects use in enforcing the TSP*].**

Application Note: The DAC and audit policies require that each subject acting on behalf of a user has a user identity associated with the subject. While this identity is typically the one used at the time of identification to the system, the DAC policy enforced by the TSF may include provisions for making access decisions based upon a different user identity, such as the "set user ID (su)"

command in UNIX.

For the assignment, in order to be compliant to this PP the ST author must include all attributes that are used in enforcing the security policy of the TOE. While some attributes listed in the assignments for FIA_ATD.1e and f may not apply to the bound subject (e.g., # of failed login attempts) it is expected that most of the attributes listed would apply.

FIA_USB.1.2 The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users:

a) For administrative users, provide restrictive defaults for security attributes identified in FIA_ATD.1, and

Application Note: An example of implementing restrictive defaults can be done through a Least Privilege mechanism. Least privilege is the characteristic whereby an entity (e.g. subject) has only the minimum privileges (authorizations, permissions, etc.) required to function and has them only when it needs them; this helps ensure that, should something go amiss, the extent of resulting damage would be minimal.

b) Restrict the ability to specify alternative initial user security attributes (that override the default attributes) to authorized administrators.

FIA_USB.1.3 The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users:

a) User security attribute changes shall take effect at next user logon.

Application Note: While the maximum delay for the changes to take effect is on the next user logon, a better approach is for immediate enforcement (e.g., when the attribute is next invoked). Implementations that do better than "next user logon" should modify this element appropriately so that credit for this implementation can be given.

5.5 Security Management (FMT)

5.5.1 Management of Functions in TSF (FMT_MOF)

5.5.1.1 Management of Security Functions Behavior (for specification of auditable events) (FMT_MOF.1(1))

Dependencies: FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

FMT_MOF.1.1(1) **Refinement:** The TSF shall restrict the ability to <u>disable and enable</u> the **audit** functions **and** to **specify which events are to be audited (see FAU_SEL.1.1)** to **the authorized administrators.**

Application Note: To "specify" means the ability to select what events will be audited.

5.5.1.2 Management of Security Functions Behavior (for authentication data) (FMT_MOF.1(2))

Dependencies: FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

FMT_MOF.1.1(2) **Refinement:** The TSF shall restrict the ability to **manage the values of security attributes associated with user authentication data** to **authorized administrators.7**

Application Note: The word "manage" includes but is not limited to create, initialize, change default, modify, delete, clear, append, and query. The security attributes associated with user authentication data referenced by this requirement include those that are specified by FIA_AFL and FIA_SOS.

5.5.2 Management of Security Attributes (FMT_MSA)

5.5.2.1 Management of Security Attributes (for Discretionary Access Control) (FMT_MSA.1(1))

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]
FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

FMT_MSA.1.1(1) Refinement: The TSF shall enforce the **Discretionary Access Control policy** to restrict the ability to **change** the value of object security attributes to **authorized administrators, owners of the object [assignment: rules that need to be satisfied for other users to perform the operations].8**

5.5.2.2 Management of Security Attributes (for Object Ownership) (FMT_MSA.1(2))

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]
FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

FMT_MSA.1.1(2) **Refinement:** The TSF shall enforce the **Discretionary Access Control policy** to restrict the ability to **change object ownership** to **authorized administrators.9**

Application Note: This requirement prevents a user from changing object ownership to another user.

5.5.2.3 Secure Security Attributes (FMT_MSA.2)

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]
FMT_MSA.1 Management of security attributes
FMT_SMR.1 Security roles

FMT_MSA.2.1 **Refinement:** The TSF shall ensure that only **valid** values are accepted for **all security attributes**.10

Application Note: Valid implies that the values assigned to security attributes are valid with respect to the secure state and fall within an appropriate range for that attribute (e.g., the password length attribute must be a non-negative integer).

5.5.2.4 Static Attributes Initialization (FMT_MSA.3)

Dependencies: FMT_MSA.1 Management of security attributes
FMT_SMR.1 Security roles

FMT_MSA.3.1 The TSF shall enforce the **Discretionary Access Control policy** to provide restrictive default values for security attributes that are used to enforce the SFP.

Application Note: The TOE must provide protection by default for all objects at creation time. This may allow authorized users to explicitly specify the desired access controls upon the object at its creation, provided that there is no window of vulnerability through which unauthorized access may be gained to newly-created objects.

FMT_MSA.3.2 The TSF shall allow the **authorized administrator** to specify alternative initial values to override the default values when an object or information is created.

Application Note: This requirement applies as a system-wide default, However, users may be allowed to define default values for objects they create (e.g., per user or per object type).

5.5.3 Management of TSF Data (FMT_MTD)

5.5.3.1 Management of TSF Data (for general TSF data) (FMT_MTD.1(1))

Dependencies: FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

FMT_MTD.1.1(1) The TSF shall restrict the ability to **manage** the **TSF data except for audit records, user security attributes, authentication data, and critical cryptographic security parameters** to **authorized administrators**.

Application Note: The word "manage" includes but is not limited to create, initialize, change default, modify, delete, clear, append, and query. Security attributes associated with user authentication data include password length, password expiration, password history, etc. The restrictions for audit records, user security attributes, authentication data, and critical cryptographic security parameters are specified below.

5.5.3.2 Management of TSF Data (for audit data) (FMT_MTD.1(2))

Dependencies: FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

FMT_MTD.1.1(2) The TSF shall restrict the ability to query, delete, and clear the **audit records** to **authorized administrators**.

Application Note: This requirement applies to actions taken on the entire audit file/log, not actions on individual audit records.

5.5.3.3 Management of TSF Data (for initialization of user security attributes) (FMT_MTD.1(3))

Dependencies: FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

FMT_MTD.1.1(3) The TSF shall restrict the ability to **initialize user security attributes** to **authorized administrators**.

5.5.3.4 Management of TSF Data (for modification of user security attributes, other than authentication data) (FMT_MTD.1(4))

Dependencies: FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

FMT_MTD.1.1(4) The TSF shall restrict the ability to modify **user security attributes, other than authentication data,** to **authorized administrators.**

5.5.3.5 Management of TSF Data (for modification of authentication data) (FMT_MTD.1(5))

Dependencies: FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

FMT_MTD.1.1(5) The TSF shall restrict the ability to modify **authentication data** to **authorized administrators and users modifying their own authentication data.**

5.5.3.6 Management of TSF Data (for reading of authentication data) (FMT_MTD.1(6))

Dependencies: FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

FMT_MTD.1.1(6) **Refinement:** The TSF shall **prevent reading of authentication data**.[11]

5.5.3.7 Management of TSF Data (for critical cryptographic security parameters) (FMT_MTD.1(7))

Dependencies: FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

FMT_MTD.1.1(7) The TSF shall restrict the ability to **manage** the **critical cryptographic security parameters and data related to cryptographic configuration** to **authorized administrators**.

Application Note: The word "manage" includes but is not limited to create, initialize, change default, modify, delete, clear, append, and query. Critical cryptographic security parameters are defined in the glossary where examples are also provided. Examples of data related to cryptographic configuration include, but are not limited to: setting of the cryptographic algorithm, setting the cryptographic mode of operation, setting the key length, setting a hash digest size, etc."

5.5.4 Revocation (FMT_REV)

5.5.4.1 Revocation (to authorized administrators) (FMT_REV.1(1))

Dependencies: FMT_SMR.1 Security roles

FMT_REV.1.1(1) The TSF shall restrict the ability to revoke **security attributes** associated with the users under the control of the TSF to **authorized administrators**.

Application Note: The phrase "revoke security attributes" means to change attributes so that access is revoked.

FMT_REV.1.2(1) **Refinement:** The TSF shall enforce the **revocation of security-relevant authorizations at the next logon.**[12]

Application Note: Security-relevant authorizations include the ability of authorized users to log in or perform privileged operations. An example of revoking a security-relevant authorization is the deletion of a user account upon which system access is immediately terminated.

5.5.4.2 Revocation (to owners and authorized administrators) (FMT_REV.1(2))

Dependencies: FMT_SMR.1 Security roles

FMT_REV.1.1 (2) **Refinement:** The TSF shall restrict the ability to revoke **security attributes of named objects** to **owners of the named object and authorized administrators**.13

Application Note: The term "revoke security attributes" means "change attributes so that access is revoked".

FMT_REV.1.2 (2) **Refinement:** The TSF shall enforce the **revocation of access rights associated with named objects when an access check is made.**14

Application Note: The state where access checks are made determines when the access control policy enforces revocation. The access control policy may include immediate or delayed revocation. The access rights are considered to have been revoked when all subsequent access control decisions made by the TSF use the new access control information. In cases where a previous access control decision was made to permit an operation, it is not required that every subsequent operation make an explicit access control decision.

5.5.5 Security Attribute Expiration (FMT_SAE)

5.5.5.1 Time-limited authorization (FMT_SAE.1)

Dependencies: FMT_SMR.1 Security roles
FPT_STM.1 Reliable time stamps

FMT_SAE.1.1 The TSF shall restrict the capability to specify an expiration time for **authorized user authentication data** to **the authorized administrator**.

FMT_SAE.1.2 **Refinement:** The TSF shall be able to **force the associated authorized user to change their authentication information prior to being able to successfully log on** after the expiration time has passed. **15**

5.5.6 Specification of Management Functions (FMT_SMF)

5.5.6.1 Specification of Management Functions (FMT_SMF.1)

Dependencies: No dependencies.

FMT_SMF.1.1 The TSF shall be capable of performing the following security management functions: **all security management functions identified in other sections of this PP**.

Application Note: The security management functions for FMT_SMF.1 are distributed throughout the PP and are included as part of the requirements in FMT_MOF, FMT_MSA, FMT_MTD, FMT_REV, FMT_SAE, FPT_TST, FRU_RSA and any cryptographic management functions specified in the reference standards. Compliance to these requirements satisfies compliance with FPT_SMF.1.

5.5.7 Security Management Roles (FMT_SMR)

5.5.7.1 Security Roles (FMT_SMR.1)

Dependencies: FIA_UID.1 Timing of identification

FMT_SMR.1.1 The TSF shall maintain the roles:

a) authorized administrator,

Application Note: Any user that is authorized to modify the TOE such that the DAC policy is bypassed is by definition, an authorized administrator. The TOE may provide multiple administrator roles (audit administrator, security administrator, etc).

b) [*assignment: at least one other mutually exclusive role that is derived from a proper subset of the above role*].

Application Note: At least one additional role must be defined; multiple additional roles are allowed as well. All additional roles must be distinct from each other and from the authorized administrator role. The associated requirements (for example, FMT_MTD.1(x)) must be appropriately refined such that each role is mutually exclusive from all other roles. For example, creating an audit administrator role from a subset of the authorized administrator role would require refining all requirements related to audit (e.g., FMT_MTD.1.1(2)) to state "audit administrator" vice "authorized administrator".

FMT_SMR.1.2 **Refinement:** The TSF shall be able to associate **authorized** users with roles.

5.6 Protection of the TOE Security Functions (FPT)

5.6.1 Internal TOE TSF Data Transfer (FPT_ITT)

5.6.1.1 Basic Internal TSF Data Transfer Protection (FPT_ITT.1)

Dependencies: FCS_COP.1 Cryptographic operation

FPT_ITT.1.1 **Refinement:** The TSF shall protect TSF data from disclosure when it is transmitted between separate parts of the TOE **through the use of the TSF-provided cryptographic services: [assignment: *FCS_COP-specified service used to protect TSF data from disclosure]***.

Application Note: The ST author includes a reference to the applicable cryptographic service into the assignment statement (e.g., if AES is used, then referring to FCS_COP.1(1) would be sufficient).

5.6.1.2 TSF Data Integrity Monitoring (FPT_ITT.3)

Dependencies: FPT_ITT.1 Basic internal TSF data transfer protection
FCS_COP.1 Cryptographic operation

FPT_ITT.3.1 **Refinement:** The TSF shall be able to detect **modification and insertion of** TSF data transmitted between separate parts of the TOE **through the use of the TSF-provided cryptographic services: [assignment: *FCS_COP-specified service used to provide modification detection]***

Application Note: The use of a cryptographic signature over the transmitted TSF data is an example of a valid implementation. The ST author includes a reference to the applicable cryptographic service into the assignment statement.

FPT_ITT.3.2 Upon detection of a data integrity error, the TSF shall take the following actions:

a) audit event, and

b) [*assignment: specify the action to be taken*].

Application Note: Additional actions ST author might consider are: retransmission of data and, an alarm after reaching a retransmission threshold.

5.6.2 Trusted Recovery (FPT_RCV)

5.6.2.1 Manual Recovery (FPT_RCV.1)

Dependencies: AGD_OPE.1 Operational user guidance

FPT_RCV.1.1 **Refinement:** After **a failure or service discontinuity that may lead to a violation of the TSP**, the TSF shall enter a maintenance mode where the ability to return **the TOE** to a secure state is provided.

Application Note: In maintenance mode normal operation might be impossible or severely restricted, as otherwise insecure situations might occur. Typically, only authorized users should be allowed access to this mode.

5.6.3 Time Stamps (FPT_STM)

5.6.3.1 Reliable Time Stamps (FPT_STM.1)

Dependencies: No dependencies.

FPT_STM.1.1 The TSF shall be able to provide reliable time stamps.

Application Note: A time stamp includes the correct date and time such that the order of events can be determined.

5.6.4 Internal TOE TSF Data Replication Consistency (FPT_TRC)

5.6.4.1 Extended: Internal TSF Data Consistency (FPT_TRC_EXT.1)

Dependencies: No dependencies.

FPT_TRC_EXT.1.1 The TSF shall ensure that TSF data is consistent between parts of the TOE by providing a mechanism to bring inconsistent TSF data into a consistent state without undue delay.

Application Note: In general, it is impossible to achieve complete, constant consistency of TSF data that is distributed to remote portions of a TOE because distributed portions of the TSF may be active at different times or disconnected from one another. This requirement attempts to address this situation in a practical manner by acknowledging that there will be TSF data inconsistencies but that they will be corrected without undue delay. For example, a TSF could provide timely consistency through periodic broadcast of TSF data to all TSF nodes maintaining replicated TSF data. Another example approach is for the TSF to provide a mechanism to explicitly probe remote TSF nodes for inconsistencies and respond with action to correct the identified inconsistencies.

5.6.5 TSF Self Test (FPT_TST)

5.6.5.1 Extended: TSF Testing (FPT_TST_EXT.1)

Dependencies: FCS_COP.1 Cryptographic operation
FCS_RBG_EXT.1 Random number generation

FPT_TST_EXT.1.1 The TSF shall run a suite of self tests in accordance with FIPS PUB 140-2 during initial start-up (on power on) to demonstrate the correct operation of the cryptographic modules**.**

Application Note: Here, "start-up" refers to start-up of the cryptomodule, and not necessarily start-up of the TSF.

FPT_TST_EXT.1.2 The TSF shall provide the capability to verify the integrity of stored TSF executable code when it is loaded for execution through the use of the TSF-provided cryptographic services.

Application Note: Refer to FCS_COP.1.1(2) and FCS_COP.1.1(3) for TSF-provided cryptographic services .

FPT_TST_EXT.1.3 The TSF shall verify the integrity of the following TSF data: authentication data, [assignment: *other TSF data to be verified*] at start up.

Application Note: In the assignment, the ST author should list other TSF data on which to apply the integrity. These data should be chosen based on the SFRs included in the ST, and cover access control permissions (FDP_ACF/FDC_IFC), security attributes associated with users (FIA_ATD), etc.

5.7 Resource Utilization (FRU)

5.7.1 Resource Allocation (FRU_RSA)

5.7.1.1 Maximum Quotas (FRU_RSA.1)

Dependencies: No dependencies.

FRU_RSA.1.1(1) The TSF shall enforce maximum quotas of the following resources: **portion of shared persistent storage** that individual **authorized** users can use simultaneously.

Application Note: For persistent storage, simultaneously means that the shared media contains data belonging to more than one user.

5.8 TOE Access (FTA)

5.8.1 Limitation on multiple concurrent sessions (FTA_MCS)

5.8.1.1 Basic limitation on multiple concurrent sessions (FTA_MCS.1)

Dependencies: FIA_UID.1 Timing of identification

FTA_MCS.1.1 **Refinement:** The TSF shall **enforce a** maximum number of concurrent **interactive** sessions per user.**16**

FTA_MCS.1.2 **Refinement:** The TSF shall allow **an authorized administrator to set the maximum number of concurrent interactive** sessions per user.**17**

Application Note: In distributed TOE implementations where synchronization of TSF data is a concern, the internal TSF data consistency requirement FPT_TRC_EXT.1 applies and any violations of the above requirement must be remedied at every synchronization.

5.8.2 Session Locking (FTA_SSL)

5.8.2.1 TSF-Initiated Session Locking (FTA_SSL.1)

Dependencies: FIA_UAU.1 Timing of authentication

FTA_SSL.1.1 The TSF shall lock an interactive session after **an authorized administrator specified time interval of user inactivity** by:

a) clearing or overwriting display devices, making the current contents unreadable.

b) disabling any activity of the user's data access/display devices other than unlocking the session.

FTA_SSL.1.2 **Refinement:** The TSF shall require the **user to re-authenticate** to unlock the session.[18]

5.8.2.2 User-Initiated Locking (FTA_SSL.2)

Dependencies: FIA_UAU.1 Timing of authentication

FTA_SSL.2.1 The TSF shall allow user-initiated locking of the user's own interactive session by:

a) clearing or overwriting display devices, making the current contents unreadable.

b) disabling any activity of the user's data access/display devices other than unlocking the session.

FTA_SSL.2.2 **Refinement:** The TSF shall require the **user to re-authenticate** to unlock the session.[19]

5.8.3 TOE Access Banners (FTA_TAB)

5.8.3.1 Default TOE access banners (FTA_TAB.1)

Dependencies: No dependencies.

FTA_TAB.1.1 **Refinement:** Before establishing a user session, the TSF shall display an **authorized-administrator specified** advisory **notice and consent** warning message regarding unauthorized use of the TOE.

Application Note: It should be noted that not all interactions with the TSF will require an access banner to be displayed. The requirement should be interpreted to cover only sessions initiated by a human.

5.8.4 TOE Access History (FTA_TAH)

5.8.4.1 TOE Access History (FTA_TAH.1)

Dependencies: No dependencies.

FTA_TAH.1.1 **Refinement:** Upon successful **interactive** session establishment, the TSF shall display **to the authorized user** the date and time of **that authorized user's** last successful **interactive** session establishment.

FTA_TAH.1.2 **Refinement:** Upon successful **interactive** session establishment, the TSF shall display **to the authorized user** the date and time of the last unsuccessful attempt and the number of unsuccessful attempts at **interactive** session establishment **for that user identifier** since the last successful **interactive** session establishment.

Application Note: In the above elements, for distributed systems, the date, time, and number of

failed attempts need to be accurate to the degree that results when implementing FPT_TRC_EXT.1.

FTA_TAH.1.3 **Refinement:** The TSF shall not erase the access history information from the **authorized** user interface without giving the **authorized** user the opportunity to review the information.

End Notes

This section records the functional requirements where deletions of Common Criteria text were performed.

---

1 A deletion of CC text was performed in FAU_SAR.1.2. Rationale: The word "user" was replaced with "authorized administrator". By default, authorized administrators are the only users with read access to audit records unless granted explicit read-access (FAU_SAR.2). The words “using a tool to access the audit records” was added for clarity.

FAU_SAR.1.2 **Refinement:** The TSF shall provide the audit records in a manner suitable for the ~~user~~ **authorized administrator** to interpret the information **using a tool to access the audit records**.

2 A deletion of CC text was performed in FAU_SAR.3.1. Rationale: The word “apply” was replaced with “perform” to make the requirement clearer.

FAU_SAR.3.1 **Refinement:** The TSF shall provide the ability to ~~apply~~ **perform searches and sorting** of audit data based on **the following attributes:**

3 A deletion of CC text was performed in FAU_SEL.1.1. Rationale: To make the requirement clearer, the words “select the set of audited” were replaced with “include or exclude auditable” and the word “auditable” was replaced with “audited”.

FAU_SEL.1.1 **Refinement:** The TSF shall be able to ~~select the set of audited~~ **include or exclude auditable** events from the set of all ~~auditable~~ **audited** events based on the following attributes:

4 A deletion of CC text was performed in FAU_STG.1.2. Rationale: The word “unauthorized” was deleted, since no one can be authorized to modify audit records.

FAU_STG.1.2 **Refinement:** The TSF shall be able to prevent ~~unauthorized~~ modifications to the stored audit records in the audit trail.

5 A deletion of CC text was performed in FDP_ACF.1.2. Rationale: The word “controlled” was deleted because there is no need to specify that subjects and objects are controlled.

FDP_ACF.1.2 **Refinement:** The TSF shall enforce the following rules to determine if an operation among ~~controlled~~ subjects and ~~controlled~~ named objects is …

6 A deletion of CC text was performed in FIA_UAU.6.1. Rationale: The words “under the conditions” were deleted for better clarity and flow on the element.

FIA_UAU.6.1 **Refinement:** The TSF shall re-authenticate the user ~~under the conditions~~ **when changing authentication data**.

7 A deletion of CC text was performed in FMT_MOF.1.1(2). Rationale: The selection [selection: *determine the behavior of, disable, enable, modify the behavior of*] and the words "the functions" were deleted and replaced with a better wording to ensure that the specific management of authentication functions were clearly conveyed.

FMT_MOF.1.1(2) **Refinement:** The TSF shall restrict the ability to **manage** ~~[selection: determine the behavior of, disable, enable, modify the behavior of] the functions~~ **the values of security attributes associated with user authentication data** to **authorized administrators**.

8 A deletion of CC text was performed in FMT_MSA.1.1(1). Rationale: The assignment “[assignment: list of security attributes]" was deleted for clarity and better flow of the requirement. The requirement is intended to restrict any changes to any of the values of all object security attributes to authorized administrators and object owners.

FMT_MSA.1.1(1) **Refinement:** The TSF shall enforce the **Discretionary Access Control policy** to restrict the

ability to **change** the **value of the object** security attributes ~~[assignment: list of security attributes]~~ to **authorized administrators and owners of the object**.

9 A deletion of CC text was performed in FMT_MSA.1.1(2). Rationale: The assignment "[assignment: list of security attributes]" was deleted for clarity and better flow of the requirement. The requirement is intended to further restrict changes to object ownership to only authorized administrators.

FMT_MSA.1.1(2) **Refinement:** The TSF shall enforce the **Discretionary Access Control policy** to restrict the ability to **change object ownership** ~~[assignment: list of security attributes]~~ to **authorized administrators and owners of the object**.

10 A deletion of CC text was performed in FMT_MSA.2.1. Rationale: The word "secure" was deleted and replaced with "valid" for clarity and better flow of the requirement.

FMT_MSA.2.1 **Refinement**: The TSF shall ensure that only ~~secure~~ **valid** values are accepted for **all security attributes**.

11 A deletion of CC text was performed in FMT_MTD.1.1(6). Rationale: The words "restrict the ability to" was replaced with "prevent" and the assignment "to [assignment: the authorized identified roles]." was deleted for clarity and better flow of the requirement.

FMT_MTD.1.1(6) **Refinement:** The TSF shall **prevent** ~~restrict the ability to~~ **reading of** **authentication data** ~~to [assignment: the authorized identified roles]~~.

12 A deletion of CC text was performed in FMT_REV.1.2 (1). Rationale: The word "rules" was deleted for clarity and better flow of the requirement.

FMT_REV.1.2(1) **Refinement:** The TSF shall enforce the ~~rules~~ **immediate revocation of security-relevant authorizations**.

13 A deletion of CC text was performed in FMT_REV.1.1 (2). Rationale: The words "associated with the" and "under the control of the TSF" were deleted for clarity and better flow of the requirement.

FMT_REV.1.1 (2) **Refinement:** The TSF shall restrict the ability to revoke **security attributes** ~~associated with the~~ **of named objects** ~~under the control of the TSF~~ to **owners of the named object and authorized administrators.**

14 A deletion of CC text was performed in FMT_REV.1.2 (2). Rationale: The word "rules" was deleted for clarity and better flow of the requirement.

FMT_REV.1.2 (2) **Refinement:** The TSF shall enforce the ~~rules~~ **revocation of access rights associated with named objects when an access check is made**.

15 A deletion of CC text was performed in FMT_SAE.1.2. Rationale: The words "For each of these security attributes," and "for the indicated security attribute" were deleted for clarity and better flow of the requirement.

FMT_SAE.1.2 **Refinement:** ~~For each of these security attributes~~, The TSF shall be able to **lock out the associated authorized user account** after the expiration time ~~for the indicated security attribute~~ has passed.

16 A deletion of CC text was performed in FTA_MCS.1.1. Rationale: The words "restrict the" were replaced with "enforce a" and the words "that belong to the same" were deleted for clarity and better flow of the requirement.

FTA_MCS.1.1 **Refinement:** The TSF shall ~~restrict the~~ **enforce a** maximum number of concurrent **interactive** sessions ~~that belong to the same~~ per user.

17 A deletion of CC text was performed in FTA_MCS.1.2. Rationale: The words "enforce, by default, a limit of" were deleted to refine the requirement to allow for a settable limit of sessions per user.

FTA_MCS.1.2 **Refinement:** The TSF shall ~~enforce, by default, a limit of~~ allow **an administrator to set the maximum number of concurrent interactive** sessions per user.

18 A deletion of CC text was performed in FTA_SSL.1.2. Rationale: The words "following events to occur" were deleted for clarity and better flow of the requirement.

FTA_SSL.1.2 **Refinement:** The TSF shall require the ~~following events to occur~~ **user to re-authenticate** ~~prior~~ to unlock~~ing~~ the session.

19 A deletion of CC text was performed in FTA_SSL.2.2. Rationale: The words "following events to occur" were deleted for clarity and better flow of the requirement.

FTA_SSL.2.2 **Refinement:** The TSF shall require the ~~following events to occur~~ **user to re-authenticate** ~~prior~~ to unlock~~ing~~ the session.

6. Security Assurance Requirements

The TOE security assurance requirements summarized in Table 1 identify the management and evaluative activities required to address the threats and policies identified in section 3 of this protection profile. Section 7.6 provides a justification for the chosen security assurance requirements and the selected assurance level EAL2 augmented with ALC_FLR.2 (Flaw Remediation).

Table 1: TOE Assurance Requirements

Assurance Class	Assurance Components	Assurance Components Description
Development	ADV_ARC.1	Security Architecture Description
	ADV_FSP.2	Security-enforcing Functional Specification
	ADV_TDS.1	Basic design
Guidance Documents	AGD_OPE.1	Operational user guidance
	AGD_PRE.1	Preparative procedures
Life Cycle Support	ALC_CMC.2	Use of a CM system
	ALC_CMS.2	Parts of the TOE CM coverage
	ALC_DEL.1	Delivery procedures
	ALC_FLR.2	**Flaw Reporting Procedures**
Tests	ATE_COV.1	Evidence of coverage
	ATE_FUN.1	Functional testing
	ATE_IND.2	Independent testing - sample
Vulnerability Assessment	AVA_VAN.2	Vulnerability analysis

6.1 Development (ADV)

6.1.1 Security Architecture (ADV_ARC)

6.1.1.1 Security architecture description (ADV_ARC.1)

Dependencies: ADV_FSP.1 Basic functional specification
ADV_TDS.1 Basic design

Developer action elements:

ADV_ARC.1.1D The developer shall design and implement the TOE so that the security features of the TSF cannot be bypassed.

ADV_ARC.1.2D The developer shall design and implement the TSF so that it is able to protect itself from tampering by untrusted active entities.

ADV_ARC.1.3D The developer shall provide a security architecture description of the TSF.

Content and presentation elements:

ADV_ARC.1.1C The security architecture description shall be at a level of detail commensurate with the description of the SFR-enforcing abstractions described in the TOE design document.

ADV_ARC.1.2C The security architecture description shall describe the security domains maintained by the TSF consistently with the SFRs.

ADV_ARC.1.3C The security architecture description shall describe how the TSF initialization process is secure.

ADV_ARC.1.4C The security architecture description shall demonstrate that the TSF protects itself from tampering.

ADV_ARC.1.5C The security architecture description shall demonstrate that the TSF prevents bypass of the SFR-enforcing functionality.

Evaluator action elements:

ADV_ARC.1.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

6.1.1.2 Security-enforcing functional specification (ADV_FSP.2)

Dependencies: ADV_TDS.1 Basic design

Developer action elements:

ADV_FSP.2.1D The developer shall provide a functional specification.

ADV_FSP.2.2D The developer shall provide a tracing from the functional specification to the SFRs.

Content and presentation elements:

ADV_FSP.2.1C The functional specification shall completely represent the TSF.

ADV_FSP.2.2C The functional specification shall describe the purpose and method of use for all TSFI.

ADV_FSP.2.3C The functional specification shall identify and describe all parameters associated with each TSFI.

ADV_FSP.2.4C For each SFR-enforcing TSFI, the functional specification shall describe the SFR-enforcing actions associated with the TSFI.

ADV_FSP.2.5C For SFR-enforcing TSFIs, the functional specification shall describe direct error

messages resulting from processing associated with the SFR-enforcing actions.

ADV_FSP.2.6C The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

Evaluator action elements:

ADV_FSP.2.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

ADV_FSP.2.2E The evaluator *shall determine* that the functional specification is an accurate and complete instantiation of the SFRs.

6.1.1.3 Basic design (ADV_TDS.1)

Dependencies: ADV_FSP.2 Security-enforcing functional specification

Developer action elements:

ADV_TDS.1.1D The developer shall provide the design of the TOE.

ADV_TDS.1.2D The developer shall provide a mapping from the TSFI of the functional specification to the lowest level of decomposition available in the TOE design.

Content and presentation elements:

ADV_TDS.1.1C The design shall describe the structure of the TOE in terms of subsystems.

ADV_TDS.1.2C The design shall identify all subsystems of the TSF.

ADV_TDS.1.3C The design shall describe the behavior of each SFR-supporting or SFR-non-interfering TSF subsystem in sufficient detail to determine that it is not SFR-enforcing.

ADV_TDS.1.4C The design shall summarize the SFR-enforcing behavior of the SFR-enforcing subsystems.

ADV_TDS.1.5C The design shall provide a description of the interactions among SFR-enforcing subsystems of the TSF, and between the SFR-enforcing subsystems of the TSF and other subsystems of the TSF.

ADV_TDS.1.6C The mapping shall demonstrate that all behavior described in the TOE design is mapped to the TSFIs that invoke it.

Evaluator action elements:

ADV_TDS.1.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

ADV_TDS.1.2E The evaluator *shall determine* that the design is an accurate and complete instantiation of all security functional requirements.

6.2 Guidance Documents (AGD)

6.2.1 Operational User Guidance (ADG_OPE)

6.2.1.1 Operational user guidance (AGD_OPE.1)

Dependencies: ADV_FSP.1 Basic functional specification

Developer action elements:

AGD_OPE.1.1D The developer shall provide operational user guidance.

Content and presentation elements:

AGD_OPE.1.1C 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.

AGD_OPE.1.2C The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.

AGD_OPE.1.3C 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.

AGD_OPE.1.4C 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.1.5C The operational user guidance shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AGD_OPE.1.6C 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.1.7C The operational user guidance shall be clear and reasonable.

Evaluator action elements:

AGD_OPE.1.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

6.2.2 Preparative Procedures (AGD_PRE)

6.2.2.1 Preparative procedures (AGD_PRE.1)

Dependencies: No dependencies.

Developer action elements:

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

Content and presentation elements:

AGD_PRE.1.1C The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.

AGD_PRE.1.2C The preparative procedures shall describe all the steps necessary for secure installation of the TOE 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.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AGD_PRE.1.2E The evaluator shall apply the preparative procedures to confirm that the TOE can be prepared securely for operation.

6.3 Life-cycle Support (ALC)

6.3.1 CM Capabilities (ALC_CMC)

6.3.1.1 Use of a CM system (ALC_CMC.2)

Dependencies: ALC_CMS.1 TOE CM coverage

Developer action elements:

ALC_CMC.2.1D The developer shall provide the TOE and a reference for the TOE.

ALC_CMC.2.2D The developer shall provide the CM documentation.

ALC_CMC.2.3D The developer shall use a CM system.

Content and presentation elements:

ALC_CMC.2.1C The TOE shall be labeled with its unique reference.

ALC_CMC.2.2C The CM documentation shall describe the method used to uniquely identify the configuration items.

ALC_CMC.2.3C The CM system shall uniquely identify all configuration items.

Evaluator action elements:

ALC_CMC.2.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

6.3.2 CM Scope (ALC_CMS)

6.3.2.1 Parts of the TOE CM coverage (ALC_CMS.2)

Dependencies: No dependencies.

Developer action elements:

ALC_CMS.2.1D The developer shall provide a configuration list for the TOE.

Content and presentation elements:

ALC_CMS.2.1C The configuration list shall include the following: the TOE itself; the evaluation evidence required by the SARs; and the parts that comprise the TOE.

ALC_CMS.2.2C The configuration list shall uniquely identify the configuration items.

ALC_CMS.2.3C For each TSF relevant configuration item, the configuration list shall indicate the developer of the item.

Evaluator action elements:

ALC_CMS.2.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

6.3.3 Delivery (ALC_DEL)

6.3.3.1 Delivery procedures (ALC_DEL.1)

Dependencies: No dependencies.

Developer action elements:

ALC_DEL.1.1D The developer shall document procedures for delivery of the TOE or parts of it to the consumer.

ALC_DEL.1.2D The developer shall use the delivery procedures.

Content and presentation elements:

ALC_DEL.1.1C The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to the consumer.

Evaluator action elements:

ALC_DEL.1.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

6.3.4 Flaw Remediation (ALC_FLR)

6.3.4.1 Flaw reporting procedures (ALC_FLR.2)

Dependencies: No dependencies.

Developer action elements:

ALC_FLR.2.1D The developer shall document flaw remediation procedures addressed to TOE

developers.

ALC_FLR.2.2D The developer shall establish a procedure for accepting and acting upon all reports of security flaws and requests for corrections to those flaws.

ALC_FLR.2.3D The developer shall provide flaw remediation guidance addressed to TOE users.

Content and presentation elements:

ALC_FLR.2.1C The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.

ALC_FLR.2.2C The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.

ALC_FLR.2.3C The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.

ALC_FLR.2.4C The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users.

ALC_FLR.2.5C The flaw remediation procedures shall describe a means by which the developer receives from TOE users reports and enquiries of suspected security flaws in the TOE.

ALC_FLR.2.6C The procedures for processing reported security flaws shall ensure that any reported flaws are remediated and the remediation procedures issued to TOE users.

ALC_FLR.2.7C The procedures for processing reported security flaws shall provide safeguards that any corrections to these security flaws do not introduce any new flaws.

ALC_FLR.2.8C The flaw remediation guidance shall describe a means by which TOE users report to the developer any suspected security flaws in the TOE.

Evaluator action elements:

ALC_FLR.2.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

6.4 Tests (ATE)

6.4.1 Coverage (ATE_COV)

6.4.1.1 Evidence of coverage (ATE_COV.1)

Dependencies: ADV_FSP.2 Security-enforcing functional specification
ATE_FUN.1 Functional testing

Developer action elements:

ATE_COV.1.1D The developer shall provide evidence of the test coverage.

Content and presentation elements:

ATE_COV.1.1C The evidence of the test coverage shall show the correspondence between the tests in the test documentation and the TSFIs in the functional specification.

Evaluator action elements:

ATE_COV.1.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

6.4.2 Functional Tests (ATE_FUN)

6.4.2.1 Functional testing (ATE_FUN.1)

Dependencies: ATE_COV.1 Evidence of coverage

Developer action elements:

ATE_FUN.1.1D The developer shall test the TSF and document the results.

ATE_FUN.1.2D The developer shall provide test documentation.

Content and presentation elements:

ATE_FUN.1.1C The test documentation shall consist of test plans, expected test results and actual test results.

ATE_FUN.1.2C The test plans shall identify the tests to be performed and describe the scenarios for performing each test. These scenarios shall include any ordering dependencies on the results of other tests.

ATE_FUN.1.3C The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE_FUN.1.4C The actual test results shall be consistent with the expected test results.

Evaluator action elements:

ATE_FUN.1.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

6.4.3 Independent Testing (ATE_IND)

6.4.3.1 Independent testing - sample (ATE_IND.2)

Dependencies: ADV_FSP.2 Security-enforcing functional specification
AGD_OPE.1 Operational user guidance
AGD_PRE.1 Preparative procedures
ATE_COV.1 Evidence of coverage
ATE_FUN.1 Functional testing

Developer action elements:

ATE_IND.2.1D The developer shall provide the TOE for testing.

Content and presentation elements:

ATE_IND.2.1C The TOE shall be suitable for testing.

ATE_IND.2.2C The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

Evaluator action elements:

ATE_IND.2.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.2.2E The evaluator *shall execute* a sample of tests in the test documentation to verify the developer test results.

ATE_IND.2.3E The evaluator *shall test* a subset of the TSF to confirm that the TSF operates as specified.

6.5 Vulnerability assessment (AVA)

6.5.1 Vulnerability Analysis (AVA_VAN)

6.5.1.1 Vulnerability analysis (AVA_VAN.2)

Dependencies: ADV_ARC.1 Security architecture description
ADV_FSP.1 Basic functional specification
ADV_TDS.1 Basic design
AGD_OPE.1 Operational user guidance
AGD_PRE.1 Preparative procedures

Developer action elements:

AVA_VAN.2.1D The developer shall provide the TOE for testing.

Content and presentation elements:

AVA_VAN.2.1C The TOE shall be suitable for testing.

Evaluator action elements:

AVA_VAN.2.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

AVA_VAN.2.2E The evaluator *shall perform* a search of public domain sources to identify potential vulnerabilities in the TOE.

AVA_VAN.2.3E The evaluator *shall perform* an independent vulnerability analysis of the TOE using the guidance documentation, functional specification, TOE design and security architecture description to identify potential vulnerabilities in the TOE.

AVA_VAN.2.4E The evaluator *shall conduct* penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Basic attack potential.

Application Note: The TOE version used as the basis for testing should include a reference to the specific signature set in place when this activity is conducted.

7. Rationale

This section provides the rationale for the selection, creation, and use of security objectives and requirements as defined in sections 4 and 5, respectively.

7.1 Security Objectives derived from Threats

Each of the identified threats to security is addressed by one or more security objectives. Table 7.1 below provides the mapping from security objectives to threats, as well as a rationale that discusses how the threat is addressed. Definitions are provided (*in italics*) below each threat and security objective so the PP reader can reference these without having to go back to sections 3 and 4.

Table 7.1 Mapping of Security Objectives to Threats

Threat	Objectives Addressing Threat	Rationale
T.ADMIN_ERROR *An administrator may incorrectly install or configure the TOE resulting in ineffective security mechanisms.*	O.MANAGE *The TOE will provide all the functions and facilities necessary to support the authorized administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.*	O.MANAGE contributes to mitigating this threat by providing the security mechanisms (e.g., tools for reviewing audit data) for administrators to perform TOE administration effectively, and to quickly alert the administrator of ineffective security policies on the TOE.
T.ADMIN_ROGUE *An authorized administrator's intentions may become malicious resulting in user or TSF data being compromised.*	O.ADMIN_ROLE *The TOE will provide administrator roles to isolate administrative actions.*	It is important to limit the functionality of administrative roles. If the intentions of an individual in an administrative role become malicious, O.ADMIN_ROLE mitigates this threat by isolating the administrative actions within that role and limiting the functions available to that individual. This objective presumes that separate individuals will be assigned separate distinct roles with no overlap of allowed operations between the roles.

Threat	Objectives Addressing Threat	Rationale
T.AUDIT_COMPROMISE *A malicious user or process may view audit records, cause audit records to be lost or modified, or prevent future records from being recorded, thus masking a user's actions.*	OE.PHYSICAL *Physical security will be provided for the TOE by the IT environment, commensurate with the value of the IT assets protected by the TOE.* O.AUDIT_GENERATION *The TOE will provide the capability to detect security relevant events and create records of those events in the audit trail.* O.AUDIT_PROTECTION *The TOE will provide the capability to protect audit information.* O.DOMAIN_ISOLATION *The TOE will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure.*	O.AUDIT_GENERATION provides the capability to detect and create records of security relevant events. Audit records identify the user responsible for the event and are an important form of evidence that can be used to track an attacker's actions. Tampering with or destruction of audit data by physical means is addressed by OE.PHYSICAL, which provides physical security controls to the TOE environment. O.AUDIT_PROTECTION provides the capability to specifically protect audit information from external interference, tampering, or unauthorized disclosure. O.DOMAIN_ISOLATION protects the TOE and its resources (including audit data) by ensuring that the security policies implemented by the TOE to protect the audit information are always invoked.
T.CRYPTO _COMPROMISE *A malicious user or process may cause key, data or executable code associated with the cryptographic functionality to be inappropriately accessed (viewed, modified, or deleted), thus compromising the cryptographic mechanisms and the data protected by those mechanisms.*	OE.PHYSICAL *Physical security will be provided for the TOE by the IT environment, commensurate with the value of the IT assets protected by the TOE.* O.DOMAIN_ISOLATION *The TOE will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure.*	The cryptography is afforded external protection from viewing, modification, or deletion by malicious users through physical security measures provided by the IT environment [OE.PHYSICAL]. Further, as part of the TOE's security functions (TSF), the cryptography is afforded internal protection from viewing, modification, or deletion by malicious processes and users through the domain isolation maintained by the TOE for its own execution [O.DOMAIN_ISOLATION].
T.MASQUERADE *A malicious user, process, or external IT entity may masquerade as an authorized entity to gain unauthorized access to data or TOE resources.*	O.USER_AUTHENTICATION *The TOE will verify the claimed identity of users.* O.USER_IDENTIFICATION *The TOE will uniquely identify users.*	To address this threat, O.USER_IDENTIFICATION identifies the user as a legitimate user and O.USER_AUTHENTICATION authenticates this user preventing unauthorized users, processes, or external IT entities from masquerading as an authorized entity.

Threat	Objectives Addressing Threat	Rationale
T.OPERATIONAL_ERRORS *While the TOE is operational, changes to the TOE may cause it to enter a configuration that is not able to enforce the security policies of the TOE.*	O.CORRECT_TSF_OPERATION *The TOE will provide a capability to test the TSF to ensure the correct operation of the TSF in its operational environment.*	The TOE must continue to operate correctly and enforce its security policies once it has been fielded. Some level of testing must be available to authorized users to ensure the TOE's security mechanisms continue to operate correctly once the TOE is fielded. O.CORRECT_TSF_OPERATION ensures that once the TOE is installed at a customer's location, the capability exists that the integrity of the TSF (hardware and software) can be demonstrated, and thus provides end users the confidence that the TOE's security policies continue to be enforced.
T.RESIDUAL_DATA *A user or process may gain unauthorized access to data through reallocation of TOE resources from one user or process to another.*	O.RESIDUAL_INFORMATION *The TOE will ensure that any data contained in a protected resource is not available when the resource is reallocated.*	The sharing of hardware resources such as primary and secondary storage components between users introduces the potential for information flow in violation of the TOE security policy when hardware resources are deallocated from one user and allocated to another. In order to prevent such unintended consequences, the TOE prevents the compromise of the TOE security policy through mechanisms that ensure that residual information cannot be accessed after the resource has been reallocated (O.RESIDUAL_INFORMATION). The intent here is to prevent the unauthorized flow of information that would violate the TOE security policy. The intent is not to require explicit scrubbing or overwriting of data prior to reuse of the storage resource. Therefore, the presence of "residual" data in a storage resource is acceptable as long as it cannot be accessed by subsequent users such that a violation of the TOE security policy results.
T.RESOURCE_EXHAUSTION *A malicious process or user may block others from system resources (i.e., system memory, persistent storage, and processing time) via a resource exhaustion denial of service attack.*	O.RESOURCE_EXHAUSTION *The TOE shall provide mechanisms that mitigate user attempts to exhaust persistent storage.*	The sharing of resources (i.e., persistent storage) between users introduces the potential for a malicious process or user to obstruct users from access to resources via a resource exhaustion denial-of-service attack. O.RESOURCE_EXHAUSTION mitigates this threat by requiring the TOE to provide controls to enforce maximum quotas for persistent storage.

Threat	Objectives Addressing Threat	Rationale
T.TSF_COMPROMISE *A malicious user or process may cause TSF data or executable code to be inappropriately accessed (viewed, modified, or deleted).*	OE.PHYSICAL *Physical security will be provided for the TOE by the IT environment, commensurate with the value of the IT assets protected by the TOE.* O.DOMAIN_ISOLATION *The TOE will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure.*	The tampering with or destruction of TSF hardware, software, or configuration data via physical means is addressed by the physical security controls present in the TOE environment [OE.PHYSICAL]. O.DOMAIN_ISOLATION addresses the threat of tampering with or destruction of TSF hardware, software, or configuration data by other (non-physical) means. It ensures that the TSF maintains a security domain for its own execution that protects it from interference and tampering by untrusted subjects and enforces the separation between the security domains of subjects within the TSC.
T.UNATTENDED _SESSION *A user may gain unauthorized access to an unattended session.*	O.PROTECT *The TOE will provide mechanisms to protect user data and resources.*	When an authorized user leaves an active session unattended, an unauthorized user may gain access to the unattended session. O.PROTECT mitigates this threat by providing mechanisms to protect user data and resources from unauthorized access by ensuring that the TSF will lock an interactive session and make the visible contents unreadable after a specified time interval of session inactivity.
T.UNAUTHORIZED _ACCESS *A user may gain unauthorized access (view, modify, delete) to user data.*	OE.PHYSICAL *Physical security will be provided for the TOE by the IT environment, commensurate with the value of the IT assets protected by the TOE.* O.ACCESS *The TOE will ensure that users gain only authorized access to it and to resources that it controls.* O.ACCESS_HISTORY *The TOE will display information (to authorized users) related to previous attempts to establish an interactive session.* O.PROTECT *The TOE will provide mechanisms to protect user data and resources.*	Unauthorized users may physically access TOE resources. To mitigate this threat, OE.PHYSICAL restricts the physical access only to authorized personnel. Within the computing environment, O.ACCESS restricts all access controls to authorized users based on their user identity. At the same time, O.PROTECT enforces access rules by providing mechanisms to prevent the user data from unauthorized disclosure and modification. O.ACCESS_HISTORY helps users confirm their previously established session or may help detected possible unsuccessful attempts to their account by an unauthorized user.
T.UNIDENTIFIED _ACTIONS *The administrator may fail to notice potential security violations, thus preventing the administrator from taking action against a possible security violation.*	O.AUDIT_REVIEW *The TOE will provide the capability to selectively view audit information and alert the administrator of identified potential security violations.*	The threat of an administrator failing to know about audit events may occur. To mitigate this threat, O.AUDIT_REVIEW provides the capability to selectively view audit information, and alert the administrator of identified potential security violations.

Threat	Objectives Addressing Threat	Rationale
T.UNKNOWN_STATE *When the TOE is initially started or restarted after a failure, the security state of the TOE may be unknown.*	O.RECOVERY *Procedures and/or mechanisms will be provided to assure that recovery is obtained without a protection compromise, such as from system failure or discontinuity.*	After a failure, the security condition of the TOE may be unknown. To mitigate this threat O.RECOVERY provides procedures and/or mechanisms to ensure that recovery without a protection compromise is obtained.

7.2 Objectives derived from Security Policies

Each of the identified security policies is addressed by one or more security objectives. Table 7.2 below provides the mapping from security objectives to security policies, as well as a rationale that discusses how the policy is addressed. Definitions are provided (*in italics*) below each threat and security objective so the PP reader can reference these without having to go back to sections 3 and 4.

Table 7.2 Mapping of Security Objectives to Security Policies

Security Policy	Objectives Addressing Policy	Rationale
P.ACCESS_BANNER *The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the TOE.*	O.DISPLAY_BANNER *The TOE will display (where appropriate) an advisory warning regarding use of the TOE.*	O.DISPLAY_BANNER satisfies this policy by ensuring that the TOE displays a banner that provides authorized users with an advisory warning about the unauthorized use of the TOE.
P.ACCOUNTABILITY *The users of the TOE shall be held accountable for their actions within the TOE*	O.AUDIT_GENERATION *The TOE will provide the capability to detect security relevant events and create records of those events in the audit trail.* O.AUDIT_REVIEW *The TOE will provide the capability to selectively view audit information and alert the administrator of identified potential security violations.* O.USER_IDENTIFICATION *The TOE will uniquely identify users.*	Enforcement of this policy requires that users be uniquely identified [O.USER_IDENTIFICATION] and that their security relevant actions be monitored and recorded [O.AUDIT_GENERATION]. The recorded audit information can be selectively reviewed in search of any potential security violations [O.AUDIT_REVIEW].
P.AUTHORIZATION *The TOE shall limit the extent of each user's abilities in accordance with the TSP.*	O.ACCESS *The TOE will ensure that users gain only authorized access to it and to resources that it controls.* O.PROTECT *The TOE will provide mechanisms to protect user data and resources.* O.USER_IDENTIFICATION *The TOE will uniquely identify users.*	O.ACCESS supports this policy by requiring the TOE to uniquely identify authorized users [O.USER_IDENTIFICATION] prior to allowing any TOE access or any TOE mediated access on behalf of those users. Within the TOE, O.PROTECT provides mechanisms to prevent user data from unauthorized disclosure and modification.

Security Policy	Objectives Addressing Policy	Rationale
P.AUTHORIZED_USERS *Only those users who have been authorized to access the information within the TOE may access the TOE.*	O.ACCESS *The TOE will ensure that users gain only authorized access to it and to resources that it controls.*	Within the set of all the users that may interact with the TOE, authorized users are those with access to the information within the TOE after being successfully identified and authenticated by the TOE. Access control policies are used to define the access permitted to the system and its resources. These policies are supported by the implementation of authorized user attributes that identify the user-allowed accesses to TOE information. O.ACCESS supports this policy by ensuring that users only gain authorized access to TOE information and its resources by checking user attributes before system use.
P.CRYPTOGRAPHY *The TOE shall use NIST FIPS validated cryptography as a baseline for key management (i.e., generation, access, distribution, destruction, validation and packaging, handling, and storage of keys) and for cryptographic operations (i.e., encryption, decryption, signature, hashing, key exchange, and random number generation services).*	O.CRYPTOGRAPHIC_SERVICES *The TOE will make cryptographic services available to authorized users and/or user applications.*	By building upon NIST FIPS-validated, cryptography, the TOE not only provides, but also augments the cryptographic support offered solely by baseline NIST FIPS-validated cryptography. The TOE cryptography supports key management (i.e., generation and destruction of keys) and cryptographic operations (i.e., encryption, decryption, signature, hashing, and random number generation). O.CRYPTOGRAPHIC_SERVICES provides these cryptographic services to TOE authorized users and/or user applications.
P.I_AND_A *All users must be identified and authenticated prior to accessing any controlled resources with the exception of public objects.*	O.USER_AUTHENTICATION *The TOE will verify the claimed identity of users.* O.USER_IDENTIFICATION *The TOE will uniquely identify users.*	In support of the policy to identify and authenticate a user before access is granted to any controlled resources, O.USER_IDENTIFICATION and O.USER_AUTHENTICATION will uniquely identify and authenticate the claimed authorized users.

Security Policy	Objectives Addressing Policy	Rationale
P.NEED_TO_KNOW *The TOE must limit the access to data in protected resources to those authorized users who have a need to know that data.*	O.ACCESS *The TOE will ensure that users gain only authorized access to it and to resources that it controls.* O.DISCRETIONARY_ACCESS *The TOE will control access to named objects based upon the identity of users and groups of users.* O.DISCRETIONARY_USER_CONTROL *The TOE will allow authorized users to specify the named objects may be accessed by which users and groups of users.* O.PROTECT *The TOE will provide mechanisms to protect user data and resources.*	The need-to-know policy is satisfied by the discretionary access control rules. O.DISCRETIONARY_ACCESS protects resources based on the identity of authorized users where the access to objects is directed by owners of the object [O.DISCRETIONARY_USER_CONTROL]. O.PROTECT enforces these policy rules by providing the mechanisms to protect the user data from disclosure and modifications and lastly, O.ACCESS ensures that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.
P.ROLES *The TOE shall provide multiple administrative roles for secure administration of the TOE. These roles shall be separate and distinct from each other.*	O.ADMIN_ROLE *The TOE will provide administrator roles to isolate administrative actions.*	To appropriately administer the system, O.ADMIN_ROLE requires the system to provide multiple administrator roles to isolate actions performed by these different roles. To completely satisfy this policy, separate roles must be assigned separate individuals.
P.TRACE *The TOE shall provide the ability to review the actions of individual users.*	O.AUDIT_REVIEW *The TOE will provide the capability to selectively view audit information and alert the administrator of identified potential security violations.*	A common organizational security policy is to maintain records allowing for individuals to be held responsible for the actions that they take with respect to organizational assets. Information can be one of the most valuable assets that an organization possesses. To satisfy this policy, O.AUDIT_REVIEW provides suitable mechanisms to accurately and selectively review those records by authorized personnel to provide accountability at the individual user level to determine any potential security violation.
P.TRUSTED_RECOVERY *Procedures and/or mechanisms shall be provided to assure that, after a TOE failure or other discontinuity, recovery without a protection compromise is obtained.*	O.RECOVERY *Procedures and/or mechanisms will be provided to assure that recovery is obtained without a protection compromise, such as from system failure or discontinuity.*	After a failure or other discontinuity, the security condition of the TOE may be unknown. O.RECOVERY provides procedures and/or mechanisms to ensure that recovery to a known secure state is obtained without a protection compromise.

7.3 Objectives derived from Assumptions

Each of the identified security assumptions is addressed by one or more security objectives. Table 7.3 below provides the mapping from security objectives to security policies, as well as a rationale that discusses how the policy is addressed. Definitions are provided (*in italics*) below each threat and security objective so the PP reader can reference these without having to go back to sections 3 and 4.

Table 7.3 Mapping of Security Objectives to Assumptions

Assumption	Objectives Addressing Assumption	Rationale
A.PHYSICAL *It is assumed that the IT environment provides the TOE with appropriate physical security, commensurate with the value of the IT assets protected by the TOE.*	OE.PHYSICAL *Physical security will be provided for the TOE by the IT environment, commensurate with the value of the IT assets protected by the TOE.*	Physical security must be provided for the TOE by the IT environment to ensure the TOE is capable of addressing the threats to TOE assets [OE.PHYSICAL].

7.4 Requirements Rationale

Each of the security objectives identified in sections 7.1 and 7.2 are addressed by one or more security requirements. Table 7.4 below provides the mapping from security requirements to security objectives, as well as a rationale that discusses how the security objective is met. Definitions are provided (*in italics*) below each security objective so the PP reader can reference these without having to go back to section 4.

Table 7.4 Mapping of Security Requirements to Objectives

Objectives from Policies/Threats	Requirements Meeting Objectives	Rationale
O.ACCESS *The TOE will ensure that users gain only authorized access to it and to resources that it controls.*	FDP_ACC.1 FDP_ACF.1 FIA_AFL_EXT.1 FIA_ATD.1 FMT_REV.1(1) FMT_REV.1(2) FPT_TRC_EXT.1 FTA_MCS.1 FTA_SSL.1 FTA_SSL.2	The TOE must protect itself and the resources it controls from unauthorized access. FDP_ACC.1 enforces the Discretionary Access Control (DAC) policy on all subjects and all named objects and all operations among them. The DAC policy specifies the access rules between all subjects and all named objects controlled by the TOE. While authorized users are trusted to some extent, this requirement ensures only authorized access is allowed to named objects. FDP_ACF.1 specifies the DAC policy rules that will be enforced by the TSF and determines if an operation among subjects and named objects is allowed. Furthermore, it specifies the rules to explicitly authorize or deny access to a named object based upon security attributes. FIA_AFL_EXT.1 provides a detection mechanism for unsuccessful authentication attempts. The requirement enables an authorized administrator configurable threshold that prevents unauthorized users from gaining access to authorized user's account by guessing authentication data. This mechanism prevents access by either disabling the targeted account. Thus, limiting an unauthorized user's ability to gain unauthorized access to the TOE. FIA_ATD.1 defines the attributes of users, including a userid that is used by the TOE to determine a user's identity and enforce what type of access the user has to the TOE (e.g., the TOE associates a userid with any role(s) they may assume). FMT_REV.1(1) ensures that the authorized administrator has the ability to revoke security attributes to a specific user. This revocation is immediate and helps authorized administrators control the ability of authorized users to log in or perform privileged operations. FMT_REV.1(2) ensures that the authorized administrator and owners of named objects have the ability to revoke security attributes to a specific user. This revocation occurs when an access check is made and helps authorized administrators and owners control the ability of users accessing named objects. FPT_TRC_EXT.1 ensures that the TSF data is consistent

Objectives from Policies/Threats	Requirements Meeting Objectives	Rationale
		between parts of the TOE by providing a mechanism to bring inconsistent TSF data into a consistent state in a timely manner. Such data may become inconsistent if an internal channel between parts of the TOE becomes inoperative or in the case of a distributed TOE, this can occur when parts become disabled, network connections are broken, and so on. The ability to ensure that the TSF data is consistent, between parts of the TOE, affords the TOE the ability to maintain the security policies current throughout all parts of the TOE and limits the opportunity of an outdated security policy to be enforced on parts of the TOE that may be permitting unauthorized access to the TOE and its resources. FTA_SSL.1 is used to prevent unauthorized access to the TOE and its resources when an interactive session is left unattended. This requirement ensures that the interactive session will lock by making the visible contents unreadable after a specified time interval of session inactivity. The authorized user needs to re-authenticate to unlock his session. FTA_SSL.2 is used to ensure that unauthorized access to the TOE and its resources when an interactive session is left unattended. It enables the authorized user to lock his interactive session before leaving the session unattended. This eliminates any chance for any user to acquire unauthorized access to an unattended session because there is no time interval of inactivity before the session is locked. The authorized user needs to re-authenticate to unlock his session.
O.ACCESS_HISTORY *The TOE will display information (to authorized users) related to previous attempts to establish an interactive session.*	FTA_TAH.1	FTA_TAH.1 is used to provide information about previous interactive sessions (i.e., date and time). This information is displayed to the authorized user upon each successful interactive session establishment. This requirement gives the authorized users the ability to verify their last successful interactive session and thus, is a means for determining if the previous successful interactive session establishment was authorized or not.
O.ADMIN_ROLE *The TOE will provide administrator roles to isolate administrative actions.*	FMT_SMR.1	The TOE must maintain roles to isolate administrative actions. FMT_SMR.1 ensures that a minimum of an administrative role be maintained
O.AUDIT_GENERATION *The TOE will provide the capability to detect security relevant events and create records of those events in the audit trail.*	FAU_GEN.1 FAU_GEN.2 FAU_SEL.1 FIA_USB.1 FPT_STM.1	FAU_GEN.1 defines the set of events that the TOE must be capable of recording. This requirement ensures that the authorized administrator has the ability to audit any security relevant event that takes place in the TOE. This requirement also defines the information that must be contained in the audit record for each auditable event. There is a minimum of information that must be present in every audit record and this requirement defines that, as well as the additional information that must be recorded for each auditable event. This requirement also places a requirement on the level of detail that is recorded on any additional security functional

Objectives from Policies/Threats	Requirements Meeting Objectives	Rationale
		requirements an ST author adds to this PP. FAU_GEN.2 ensures that the audit records associate a user identity with the auditable event. The association is accomplished using the userid of the authorized user. FAU_SEL.1 allows the authorized administrator to configure which auditable events will be recorded in the audit trail. This provides the administrator with the flexibility in recording only those events that are deemed necessary by site policy, thus reducing the amount of resources consumed by the audit mechanism. FIA_USB.1 plays a role is satisfying this objective by requiring a binding of security attributes associated with users that are authenticated with the subjects that represent them in the TOE. This only applies to authenticated users, since the identity of unauthenticated users cannot be confirmed. Therefore, the audit trail may not always have the proper identity of the user that causes an audit record to be generated (e.g., an attacker/user providing another user's user identifier). FPT_STM.1 ensures that the time stamps used to create the audit records are reliable. The time and date included in the time stamp is crucial when generating the audit information to ensure accountability.
O.AUDIT_PROTECTION *The TOE will provide the capability to protect audit information.*	FAU_SAR.2 FAU_STG.1	The audit trail must be protected so that only authorized users and authorized administrators may access it or delete it. FAU_SAR.2 ensures that only authorized users have read access to audit information and FAU_STG.1 ensures that audit information is not modified and protects it from unauthorized deletions.
O.AUDIT_REVIEW *The TOE will provide the capability to selectively view audit information and alert the administrator of identified potential security violations.*	FAU_SAR.1 FAU_SAR.3 FAU_STG.3	FAU_SAR.1 provides the ability for an authorized administrator to efficiently review audit records. This requirement also mandates the audit information be presented in a manner that is suitable for the administrators to interpret the audit trail. FAU_SAR.3 complements FAU_SAR.1 by providing the administrators the flexibility to specify criteria that can be used to search or sort the audit records residing in the audit trail. FAU_SAR.3 requires the administrators be able to establish the audit review criteria based on a user and identifier, date and time, so that the actions of a user can be readily identified and analyzed. Allowing the administrators to perform searches or sort the audit records based on dates, times, type of events, and success and failure of these events, provides the capability to extract the user activity to what is pertinent at that time in order facilitate the administrator's review. It is important to note that the intent of sorting in this requirement is to allow the administrators the capability to organize or group the records associated with a given criteria. FAU_STG.3 allows the authorized administrator to be alerted of the possible audit data loss if the audit trail exceeds an

Objectives from Policies/Threats	Requirements Meeting Objectives	Rationale
		authorized administrator selectable, pre-defined limit.
O.CORRECT_TSF _OPERATION *The TOE will provide a capability to test the TSF to ensure the correct operation of the TSF in its operational environment.*	FMT_MSA.2 FPT_TST_EXT.1	FMT_MSA.2. This requirement ensures that only valid values are accepted for security attributes. The values that are accepted as valid for a specific security attribute must fall within the appropriate range for that attribute (e.g., the password length attribute must be a non-negative integer). FPT_TST_EXT.1 is necessary to demonstrate the correct operation of the cryptographic algorithms and RNG/PRNG;
O.CRYPTOGRAPHIC _SERVICES *The TOE will make cryptographic services available to authorized users and/or user applications.*	FCS_BCM_EXT.1 FCS_COA_EXT.1 FCS_CKM.1(1) FCS_CKM.1(2) FCS_CKM.4 FCS_COP.1(1) FCS_COP.1(2) FCS_COP.1(3) FCS_RBG_EXT.1	Baseline cryptographic services are provided in the TOE by FIPS PUB 140-2 compliant modules [FCS_BCM_EXT.1]. Specific functional requirements in the area of cryptographic operations address data encryption and decryption [FCS_COP.1 (1)]; cryptographic signatures [FCS_COP.1 (2)]; cryptographic hashing [FCS_COP.1 (3)]; random number generation [FCS_RBG_EXT.1]; and supporting key management services [FCS_CKM.1(1), FCS_CKM.1(2), FCS_CKM.4]. These TOE requirements support cryptographic services that can be called upon by the TOE itself, or by TOE authorized users and/or user applications [FCS_COA_EXT.1].
O.DISCRETIONARY _ACCESS *The TOE will control access to named objects based upon the identity of users and groups of users.*	FDP_ACC.1 FDP_ACF.1 FIA_USB.1 FMT_MSA.3	Access to TOE resources is determined by the Discretionary Access Control policy. FDP_ACC.1 ensures that the Discretionary Access Control policy is enforced on all subjects and all named objects and all operations between them. FDP_ACF.1 defines the Discretionary Access Control rules to determine if any operation between subjects and named objects is allowed. These rules are based on the identity of the users and their group memberships. FIA_USB.1 defines the associations between user security attributes and subjects acting on behalf of that user by which policy decisions are based upon. FMT_MSA.3 ensures that the TOE provides protection by default for all named objects at creation time. This may allow authorized users to explicitly specify the desired access controls upon the object at its creation, provided that there is no window of vulnerability through which unauthorized access may be gained to newly-created objects.
O.DISCRETIONARY _USER_CONTROL *The TOE will allow authorized users to specify the named objects may be accessed by which users and groups of users.*	FMT_MSA.1(1) FMT_MSA.1(2) FMT_REV.1(2)	To allow authorized users to specify which resources may be accessed, the TOE must provide the ability for object security attributes to be changed and revoked. FMT_MSA.1(1) and FMT_MSA.1(2) restrict the ability to change the value of object security attributes to authorized administrators and owners of objects. FMT_REV.1(2) restricts the ability to revoke security attributes of named objects to authorized administrators and owners of these objects.
O.DISPLAY_BANNER	FTA_TAB.1	Before identification and authentication and the establishment

Objectives from Policies/Threats	Requirements Meeting Objectives	Rationale
The TOE will display (where appropriate) an advisory warning regarding use of the TOE.		of a user session, the TOE allows limited access by any potential users of the system in order to convey warnings and agreements for system use. Through this limited access before establishing a user session, the TSF displays an authorized, administrator-specified advisory notice and consent warning message regarding unauthorized use of the TOE [FTA_TAB.1]. In typical applications a user who continues session establishment procedures (including their successful identification and authentication) after display of the notice and warning banner effectively acknowledges the banner content and consents to the stated conditions. This banner of information can be critical in supporting legal actions related to the use of the TOE.
O.MANAGE *The TOE will provide all the functions and facilities necessary to support the authorized administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.*	FMT_MOF.1(1) FMT_MOF.1(2) FMT_MSA.1(1) FMT_MSA.1(2) FMT_MSA.3 FMT_MTD.1(1) FMT_MTD.1(2) FMT_MTD.1(3) FMT_MTD.1(4) FMT_MTD.1(5) FMT_MTD.1(7) FMT_REV.1(1) FMT_REV.1(2) FMT_SAE.1 FMT_SMF.1	In a variety of ways the TOE supports authorized administrators in the management of security functions, security attributes and data while also restricting unauthorized use. For example, the TOE provides for and restricts the following actions to authorized administrators only (except where specifically noted): • Disable and enable the audit functions, and specify which events are audited [FMT_MOF.1 (1)]. • Create, initialize, change default, modify, delete, clear, append, query, etc. the values of security attributes associated with user authentication data [FMT_MOF.1 (2)]. • Change the value of object security attributes. (Object owner is also allowed to perform this action.) [FMT_MSA.1(1), FMT_MSA.1(2)]. • Provide restrictive default values for security attributes, and specify alternative initial values to override the default values when an object or information is created. [FMT_MSA.3]. • Create, initialize, change default, modify, delete, clear, append, query, etc. the security-relevant TSF data (except audit records, user security attributes, authentication data, and critical security parameters) [FMT_MTD.1 (1)]. • Query, delete, and clear audit records [FMT_MTD.1 (2)]. • Initialize user security attributes. [FMT_MTD.1 (3)]. • Modify user security attributes, other than authentication data. [FMT_MTD.1 (4)]. • Modify authentication data. (Also allows users authorized to modify their own authentication data to do so.) [FMT_MTD.1 (5)]. • In addition, the TOE restricts the management of the critical cryptographic security parameters to an

Objectives from Policies/Threats	Requirements Meeting Objectives	Rationale
		authorized administrator [FMT_MTD.1 (7)]. • Revoke security attributes associated with the users within the TSC. [FMT_REV.1 (1)]. • Revoke security attributes of named objects within the TSC. (Object owner is also allowed to perform this action.) [FMT_REV.1 (2)]. • Specify an expiration time for authorized user authentication data. [FMT_SAE.1]. FMT_SMF.1 provides a list of the management functions specified in this PP and is required as a dependency for the management functions.
O.PROTECT *The TOE will provide mechanisms to protect user data and resources.*	FDP_ACC.1 FDP_ACF.1 FDP_RIP.2 FIA_SOS.1 FIA_UAU.7 FMT_MTD.1(6) FMT_REV.1(2)	O.PROTECT requires mechanisms be provided by the TOE to protect user data and resources. FIA_SOS.1 prescribes the metrics that must be satisfied for user authentication. If a user can't authenticate, he or she will not have the ability to access user data and resources. FIA_SOS.1 requires that the authentication mechanism provide the ability for authorized users to have a "secret" up to 16 characters in length, consisting of any combination of upper and lower case letters, numbers, and punctuation. FIA_UAU.7 ensures that no feedback that affects the ability of users to circumvent the authentication mechanism is presented during the authentication process. The TOE is allowed to provide information that would allow the user to use the authentication mechanism in a correct manner (e.g., press CTRL-ALT-DELTE, slide card quickly, center your finger and press firmly, speak louder and slowly), but not provide information that may allow alteration to their presentation that would thwart the mechanism. FMT_MTD.1(6) ensures that the authentication data is protected. No entity is allowed to read authentication data and the TSF must prevent any attempt to read it. To protect user data and resources, FDP_ACC.1, FDP_ACF.1, and FMT_REV.1(2) require a Discretionary Access policy and rules that ensures the correct access to named objects by subjects acting on behalf of users. To ensure that user data is not disclosed before a resource is reused, FDP_RIP.2 ensures that the shared memory and operating system controlled files are not available to another user thus protecting the user data.
O.RECOVERY *Procedures and/or mechanisms will be provided to assure that recovery is obtained without a protection compromise, such as from system failure or discontinuity.*	FPT_RCV.1 FPT_TRC_EXT.1	FPT_RCV.1 ensures that the system enters a maintenance mode allowing the system to be returned to a secure state after a failure or service discontinuity. In a secure state, all security policies are enforced. FPT_TRC_EXT.1 provides a mechanism to bring the TOE into a consistent state. TSF data may become inconsistent if an internal channel between parts of the TOE becomes inoperative or in the case of a distributed TOE, this can occur when parts become disabled, network connections are broken,

Objectives from Policies/Threats	Requirements Meeting Objectives	Rationale
		and so on. The ability to ensure that the TSF data is consistent, between parts of the TOE, provides the TOE the ability to maintain the security policies current throughout all parts of the TOE and limits the opportunity of an outdated security policy to be enforced on parts of the TOE that may be permitting unauthorized access to the TOE and its resources. This requirement provides the mechanisms to ensure that upon reconnection, the TSF portions will become in sync over a reasonable time period.
O.RESIDUAL _INFORMATION *The TOE will ensure that any data contained in a protected resource is not available when the resource is reallocated.*	FDP_RIP.2	FDP_RIP.2 is used to ensure the contents of resources are not available to subjects other than those explicitly granted access to the data.
O.RESOURCE_EXHAUSTION *The TOE shall provide mechanisms that mitigate user attempts to exhaust persistent storage.*	FRU_RSA.1 FTA_MCS.1	This objective requires mechanisms to prevent authorized users (or software unknowingly acting on their behalf) from exhausting important resources controlled by the TOE in a manner that adversely impacts other users or programs. TOE is required to enforce a limit on the amount of resource a given authorized user may successfully be granted. The resources that are controlled are: CPU time, disk space, system memory, and user accounts. FRU_RSA.1 is intended to enforce the notion that a single authorized user may only be allocated a "preset maximum" amount of resource. The requirement only covers persistent storage to offer confidence that entities executing on the TOE are not "starved for persistent storage" and will be allowed to initiate and complete execution. FTA_MCS.1 identifies user accounts as a system resource that could be exhausted (through multiple concurrent "logons" of a single individual). The requirement mandates that the administrator be able to limit the number of concurrent logon sessions by a single user. This ensures that a single individual could not mount a denial-of-service attack using multiple sessions as launching points. Resources (e.g., memory contained on the network card) that are not covered by the above are subject to denial of service attacks. Denial-of-service attacks of these resources should be addressed via other mechanisms such as redundant hardware.
O.DOMAIN_ISOLATION *The TOE will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure and ensures that the security policies implemented by the TOE are always invoked.*	FPT_ITT.1 FPT_ITT.3 FPT_RCV.1 FPT_TRC_EXT.1	This objective requires the protection of the TSF (and its data) from external interference, tampering or inappropriate disclosure by mandating that the TSF create and maintain a domain for its execution. Domain is defined as the logical area that the TSF provides for itself in which to operate. Common mechanisms include hardware execution domains (e.g., processor execution rings as well as other isolation mechanisms that protect TSF data when it is in transit to other TSF components.)

Objectives from Policies/Threats	Requirements Meeting Objectives	Rationale
		The requirements that implement this objective fall into two categories. The first category mandates mechanisms to implement a secure domain for execution. The second category mandates that if the TSF (for some reason) moves into an unknown or unconnected state, that it has a way to recover to a known or connected state. This ensures that the TSF can continue to protect itself even after unexpected interruptions. Requirements included in the first category are FPT_ITT.1 and FPT_ITT.3 (in addition several assurance requirements). The FPT_ITT requirements protect TSF data in transmission between remote portions of the TSF and also require that mechanisms be in place to protect against man-in-the-middle replay attacks that could attempt to interfere with the TSF policy being enforced. Requirements included in the second category are FPT_RCV.1 and FPT_TRP_EXT.1. FPT_RCV.1 is used to ensure that the TSF offers a mechanism to recover from a failed state by mandating that the TSF provide maintenance mode from which to re-initiate (or establish) a known (secure) state. This ensures that once the TSF has established a domain for its own execution it can always return to that state with confidence that this domain continues to be present. FPT_TRP_EXT.1 is used to address distributed TSFs and the fact that portions of these TSF may become disconnected over time. A disconnected portion of the TSF does not always suggest an insecure state or discontinuity of service (referenced in FPT_RCV.1). Instead, this requirement addresses the situation when a portion of a distributed TSF is disconnected from the rest of the TSF (with both pieces continuing service). Specifically, it requires that there be mechanisms provided by the TSF to ensure that upon reconnection, the TSF portions will become in sync over a reasonable time period.
O.TSF_CRYPTOGRAPHIC_INTEGRITY *The TOE will provide cryptographic integrity mechanisms for TSF data while in transit to remote parts of the TOE.*	FPT_ITT.3	This objective requires the TOE to provide cryptography that must be used to protect TSF data as it is transmitted between parts of a physically distributed TOE. FPT_ITT.3 requires that the TSF shall be able to use encryption to detect modification, insertion and replay of TSF data transmitted between separate parts of the TOE.
O.USER _AUTHENTICATION *Users must authenticate their claimed identities (see O.USER_IDENTIFICATION) before they are allowed access to the TOE.*	FIA_SOS.1 FIA_UAU.1 FIA_UAU.6 FTA_SSL.1 FTA_SSL.2	FIA_UAU.1 plays a role in satisfying this objective by ensuring that every user is authenticated before the TOE performs any TSF-mediated actions on behalf of that user. FIA_UAU.6 ensures that the authorized user changing his authentication data re-authenticates before he or she is allowed to proceed. To verify the claimed identity of an authorized user, FIA_SOS.1 prescribes the metrics that must be satisfied. It provides the mechanism that will verify the secret for user

Objectives from Policies/Threats	Requirements Meeting Objectives	Rationale
		authentication. The PP authors intentionally did not dictate that a password mechanism be required and allowed for other types of authentication mechanisms (e.g. a PIN, Token). In any case, FIA_SOS.1 requires that the authentication mechanism provide the ability for authorized users to have a "secret" up to 16 characters in length, consisting of any combination of upper and lower case letters, numbers, and punctuation FTA_SSL.1 and FTA_SSL.2 ensure that the authorized user authenticates him or herself before accessing a locked interactive session. This eliminates any chance for any user to acquire unauthorized access to an unattended session. Active interactive sessions may be locked by a user or after a specified time interval of user inactivity configured by an authorized administrator.
O.USER _IDENTIFICATION *The TOE will uniquely identify users.*	FIA_UID.1	FIA_UID.1 plays a role in satisfying this objective by ensuring that every user is identified before the TOE performs any TSF-mediated actions on behalf of that user. It also allows for the specification of a list of public objects that users are allowed read access before the user is identified.

7.5 Extended Requirements Rationale

Extended components have been included in this protection profile because the Common Criteria requirements were found to be insufficient as stated. Tables 7.5 and 7.6 include the rationale for using extended requirements.

7.5.1 Extended Functional Requirements

Table 7.5 Rationale for Extended Functional Requirements

Extended Component	Rationale
FCS_BCM_EXT.1	The CC does not provide a means of specifying a cryptographic module baseline for implementations developed in hardware, in software, or in hardware/software combinations. FCS_BCM_EXT.1 provides for the specification of the required FIPS certification based on the implementation baseline.
FCS_COA_EXT.1	FCS_COA_EXT.1 was created to require a means for applications to be able to utilize the cryptographic functionality contained in the TOE.
FCS_RBG_EXT.1	The generation of random numbers can be better stated as an explicit component to ensure that Random Number Generation (RNG) services in accordance with a FIPS-Approved RNG listed in FIPS PUB 140-2 Annex C composed and comply with the tests specified in NIST SP 800-90.

Extended Component	Rationale
FPT_TRC_EXT.1	FPT_TRC_EXT has been created to require timely consistency of replicated TSF data. Although there is a Common Criteria Requirement that attempts to address this functionality, it falls short of the needs of the environment in this protection profile. In general, it is impossible to achieve complete, constant consistency of TSF data that is distributed to remote portions of a TOE because distributed portions of the TSF may be active at different times or disconnected from one another. This requirement attempts to address this situation in a practical manner by acknowledging that there will be TSF data inconsistencies but that they will be corrected without undue delay.
FPT_TST_EXT.1	FPT_TST_EXT.1 has been created because the FPT_TST.1.2 element was removed from the original component FPT_TST.1. The element FPT_TST.1.2 states that TSF shall provide authorized users with the capability to verify the integrity of the TSF data or a subset of TSF data. This not a feasible requirement. Verifying the integrity of TSF data (e.g., passwords, session keys) is not feasible because it is constantly being updated.

7.6 Rationale for Assurance Rating

This protection profile has been developed for a U.S. Government basic robustness environment. The type of information processed by the environment establishes the need for the TOE to be evaluated at an Evaluated Assurance Level 2 Augmented (EAL2+).

8. References

[1] Common Criteria Implementation Board, Common Criteria for Information Technology Security Evaluation, CCMB-2007-09-001, Version 3.1, September 2007

[2] Department of Defense Standard, Department of Defense Trusted Computer System Evaluation Criteria (Orange Book), December 1985

[3] US Government Protection Profile for Multilevel Operating Systems in Medium Robustness Environments, Version 1.91, 16 March 2007

[4] Security Control Catalog for National Security Systems, CNSS Instruction No. 1253 (ODNI/CIO Draft v5.0) August 2008

Appendix A - Acronyms

ANSI	American National Standards Institute
CC	Common Criteria for Information Technology Security Evaluation Version 2.3
COTS	Commercial-Off-The-Shelf
DAC	Discretionary Access Control
DoD	Department of Defense
EAL	Evaluation Assurance Level
FIPS	Federal Information Processing Standard
IA	Information Assurance
IT	Information Technology
NIST	National Institute of Standards and Technology
OS	Operating System
PP	Protection Profile
RNG	Random Number Generator
SF	Security Function
SFP	Security Function Policy
SFR	Security Function Requirement
ST	Security Target
TOE	Target of Evaluation
TOM	Target of Maintenance
TSC	TSF Scope of Control
TSF	TOE Security Functions
TSFI	TSF Interface
TSP	TOE Security Policy

Appendix B - Cryptographic Standards, Policies, and Other Publications

Standards

FIPS PUB 140-2 National Institute of Standards and Technology, Security Requirements for Cryptographic Modules, Federal Information Processing Standard Publication (FIPS-PUB) 140-2, dated May 25, 2001, [http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf].

FIPS PUB 140-2 Annex C National Institute of Standards and Technology, October 2007 Approved Random Number Generators for FIPS PUB 140-2, Security Requirements for Cryptographic Modules [http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf].

FIPS PUB 180-3 National Institute of Standards and Technology, Secure Hash Standard, Federal Information Processing Standard Publication (FIPS-PUB) 180-3, dated October 2008, [http://csrc.nist.gov/publications/fips/fips180-3/fips180-3_final.pdf].

FIPS PUB 186-3 National Institute of Standards and Technology, Digital Signature Standard (DSS), Federal Information Processing Standard Publication (FIPS-PUB) 186-3, dated June 2009 [http://csrc.nist.gov/publications/fips/fips186-3/FIPS_186-3.pdf].

FIPS PUB 197 National Institute of Standards and Technology, Advanced Encryption Standard, Federal Information Processing Standard Publication (FIPS-PUB) 197, dated November 2001, [http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf].

Other Publications

NIST S.P. 800-22 National Institute of Standards and Technology Special Publication 800-22: A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications, May 2001, [http://csrc.nist.gov/publications/nistpubs/800-22/sp-800-22-051501.pdf].

NIST SP.800-56A National Institute of Standards and Technology Special Publication 800-56: Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography, March, 2007 [http://csrc.nist.gov/publications/nistpubs/800-56A/SP800-56A_Revision1_Mar08-2007.pdf].

NIST SP 800-57 National Institute of Standards and Technology Special Publication 800-57: Recommendation for Key Management, May 2006, [http://csrc.nist.gov/publications/nistpubs/800-57/sp800-57-Part1-revised2_Mar08-2007.pdf]

NIST SP 800-90 National Institute of Standards and Technology Special Publication 800-90: Recommendation for Random Number Generation Using Deterministic Random Bit Generators (Revised), March 2007, [http://csrc.nist.gov/publications/nistpubs/800-90/SP800-90revised_March2007.pdf]