

---

**HP-UX 11i v3 Common Criteria  
Security Target  
Against the  
COTS Compartmentalized Operations Protection Profile  
Operating Systems  
(CCOPP-OS)**



Prepared by: Hewlett-Packard Development Company, L.P.  
Project Manager: Mo Mohundro  
Project Document id: ST\_V1.6.doc  
Version ID: Version 1.6  
Date Prepared: October 31, 2009

# Table of Contents

Table of Contents .....	2
Table of Tables.....	5
1. ST Introduction .....	6
1.1. ST Reference .....	6
1.2. TOE Reference .....	6
1.3. TOE Overview .....	6
1.3.1. TOE Type.....	6
1.3.2. Major Security Features .....	6
1.4. TOE Description.....	7
1.4.1. TOE Physical Scope.....	7
1.4.2. TOE Logical Scope .....	8
1.4.3. Major Security Features .....	8
2. Conformance Claims.....	10
2.1. CC Conformance .....	10
2.2. PP Conformance.....	10
2.3. PP Tailoring.....	10
3. Security Problem Definition.....	11
3.1. Introduction .....	11
3.2. Threats.....	11
3.2.1. Assets .....	11
3.2.2. Threat Agents .....	11
3.2.3. Threats addressed by the TOE.....	11
3.2.4. Threats addressed by the Operational Environment .....	12
3.3. Organizational Security Policies .....	13
3.4. Assumptions .....	13
3.4.1. Usage Aspects .....	13
3.4.2. Physical Aspects.....	14
3.4.3. Personnel Aspects.....	14
4. Security Objectives .....	15
4.1. Security Objectives for the TOE .....	15
4.2. Security Objectives for the Operational Environment.....	16
4.3. Security Objectives Rationale .....	17
5. Security Requirements .....	18
5.1. Security Functional Requirements.....	18
5.1.1. Audit (FAU) .....	18
5.1.1.1. Audit Data Generation (FAU_GEN.1).....	18
5.1.1.2. User Identity Generation (FAU_GEN.2).....	20
5.1.1.3. Audit Review (FAU_SAR.1).....	20
5.1.1.4. Restricted Audit Review (FAU_SAR.2) .....	20
5.1.1.5. Selectable Audit Review (FAU_SAR.3) .....	21
5.1.1.6. Selective Audit (FAU_SEL.1).....	21
5.1.1.7. Protected Audit Trail Storage (FAU_STG.1) .....	21
5.1.1.8. Actions in Case of Possible Audit Data Loss (FAU_STG.3) .....	21
5.1.1.9. Prevention of Audit Data Loss (FAU_STG.4) .....	21
5.1.2. User Data Protection (FDP).....	22
5.1.2.1. Discretionary Access Control Policy (FDP_ACC.1-A).....	22
5.1.2.2. Discretionary Access Control Policy Rules (FDP_ACF.1-A) .....	22
5.1.2.3. Role-Based Access Control Policy (FDP_ACC.1-B).....	24
5.1.2.4. Role-Based Access Control Policy Rules (FDP_ACF.1-B) .....	24
5.1.2.5. Export of User Data (FDP_ETC.1).....	24
5.1.2.6. Mandatory Access Control Policy (FDP_IFC.1).....	24
5.1.2.7. Mandatory Access Control Policy Rules (FDP_IFF.1) .....	25

5.1.2.8.	Import of User Data (FDP_ITC.1).....	26
5.1.2.9.	Object Residual Information Protection (FDP_RIP.2) .....	26
5.1.2.10.	Subject Residual Information Protection (FDP_RIP.CCOPP) .....	26
5.1.3.	Identification and Authentication (FIA) .....	26
5.1.3.1.	Authentication Failure Handling (FIA_AFL.1).....	26
5.1.3.2.	User Attribute Definition (FIA_ATD.1).....	26
5.1.3.3.	Verification of Authentication Data (FIA_SOS.1) .....	27
5.1.3.4.	User Authentication Before Any Action (FIA_UAU.2).....	27
5.1.3.5.	Support for Multiple Authentication Mechanisms (FIA_UAU.CCOPP) .....	27
5.1.3.6.	Re-Authentication (FIA_UAU.6) .....	28
5.1.3.7.	Protected Authentication Feedback (FIA_UAU.7).....	28
5.1.3.8.	User Identification Before Any Action (FIA_UID.2).....	28
5.1.3.9.	User-Subject Binding (FIA_USB.1).....	28
5.1.4.	Security Management (FMT) .....	29
5.1.4.1.	Management of DAC Object Security Attributes (FMT_MSA.1-A) .....	29
5.1.4.2.	Management of RBAC Object Security Attributes (FMT_MSA.1-B) .....	29
5.1.4.3.	Management of Object Compartment Labels (FMT_MSA.1-C).....	30
5.1.4.4.	Management of User Security Attributes (FMT_MSA.1-D).....	30
5.1.4.5.	Secure RBAC Security Attributes (FMT_MSA.2).....	30
5.1.4.6.	DAC Static Attribute Initialization (FMT_MSA.3-A) .....	30
5.1.4.7.	RBAC Static Attribute Initialization (FMT_MSA.3-B) .....	30
5.1.4.8.	MAC Static Attribute Initialization (FMT_MSA.3-C).....	30
5.1.4.9.	Management of Audit Trail (FMT_MTD.1-A) .....	31
5.1.4.10.	Management of Audited Events (FMT_MTD.1-B).....	31
5.1.4.11.	Management of Authentication Data – Initialization (FMT_MTD.1-C).....	31
5.1.4.12.	Management of Authentication Data – Modification (FMT_MTD.1-D) .....	31
5.1.4.13.	Management of TOE Access Banner (FMT_MTD.1-E) .....	31
5.1.4.14.	Management of Role Definitions (FMT_MTD.1-F) .....	31
5.1.4.15.	Secure Role Definition Values (FMT_MTD.3).....	31
5.1.4.16.	Revocation of User Attributes (FMT_REV.1-A) .....	32
5.1.4.17.	Revocation of Object Security Attributes (FMT_REV.1-B) .....	32
5.1.4.18.	Time-Limited Authorization (FMT_SAE.1) .....	32
5.1.4.19.	Specification of Management Functions (FMT_SMF.1).....	33
5.1.4.20.	Security Roles (FMT_SMR.2).....	33
5.1.5.	Protection of the TOE Security Functions (FPT) .....	33
5.1.5.1.	Failure with Preservation of Secure State (FPT_FLS.1).....	33
5.1.5.2.	Subset Inter-TSF Confidentiality During Transmission (FPT_ITC.CCOPP).....	33
5.1.5.3.	Subset Inter-TSF Detection of Modification (FPT_ITI.CCOPP) .....	34
5.1.5.4.	Manual Recovery (FPT_RCV.1) .....	34
5.1.5.5.	Function Recovery (FPT_RCV.4).....	34
5.1.5.6.	Reliable Time Stamps (FPT_STM.1) .....	34
5.1.5.7.	Testing of External Entities (FPT_TEE.1).....	34
5.1.5.8.	TSF Testing (FPT_TST.1).....	35
5.1.6.	Resource Utilization (FRU).....	35
5.1.6.1.	Limited Priority of Service (FRU_PRS.1).....	35
5.1.6.2.	Maximum Quotas (FRU_RSA.1) .....	35
5.1.7.	TOE Access (FTA).....	35
5.1.7.1.	Limitation on Scope of Selectable Attributes (FTA_LSA.1).....	35
5.1.7.2.	Basic Limitation on Multiple Concurrent Sessions (FTA_MCS.1).....	36
5.1.7.3.	User-Initiated Termination (FTA_SSL.4) .....	36
5.1.7.4.	Default TOE Access Banners (FTA_TAB.1) .....	36
5.1.7.5.	TOE Access History (FTA_TAH.1).....	36
5.1.7.6.	TOE Session Establishment (FTA_TSE.1) .....	36
5.2.	Security Assurance Requirements .....	37
5.3.	Security Requirements Ratioanle .....	37
6.	TOE Summary Specifications .....	38

6.1.	Introduction .....	38
6.1.1.	Concepts and Terminology.....	38
6.1.1.1.	Subjects, Sessions and Privileges .....	38
6.1.1.2.	Objects.....	39
6.1.1.3.	Fine-Grained Privileges .....	40
6.1.1.4.	Discretionary Access Control (DAC) Mechanism .....	41
6.1.1.5.	Mandatory Access Control (MAC) Mechanism.....	41
6.1.1.6.	Role Based Access Control (RBAC) Mechanism .....	42
6.1.1.7.	Initial and Secure States .....	43
6.2.	Security Functions.....	43
6.2.1.	Audit (AUD).....	43
6.2.1.1.	Audit Data Collection (AUD_COLL) .....	43
6.2.1.2.	Audit Events (AUD_EVENTS).....	43
6.2.1.3.	Audit Records (AUD_RECS).....	46
6.2.1.4.	Audit Logs Viewing (AUD_VWNG).....	47
6.2.1.5.	Audit Log Files Maintenance (AUD_MTNS) .....	47
6.2.2.	Discretionary Access Control (DAC).....	48
6.2.2.1.	Discretionary Access Control for File System Objects (DAC_FS).....	48
6.2.2.2.	Discretionary Access Control for IPC Objects (DAC_IPC).....	51
6.2.3.	Mandatory Access Control (MAC) .....	52
6.2.3.1.	Mandatory Access Control for File System Objects (MAC_FS).....	52
6.2.3.2.	Mandatory Access Control for IPC Objects (MAC_IPC) .....	53
6.2.3.3.	Mandatory Access Control for Network Interface Objects (MAC_NET) .....	54
6.2.4.	Role Based Access Control (RBAC).....	54
6.2.4.1.	Role Based Access Control (RBAC) .....	54
6.2.5.	Object Reuse.....	56
6.2.5.1.	Object Reuse (OBJ_REUSE) .....	56
6.2.6.	Identification and Authentication (IA) .....	56
6.2.6.1.	User Attributes (IA_ATTR) .....	56
6.2.6.2.	User Authentication (IA_AUTH) .....	57
6.2.6.3.	User Identification (IA_UID) .....	58
6.2.6.4.	Password Selection, Generation, and Encryption (IA_PW) .....	58
6.2.7.	Session Management (SM).....	58
6.2.7.1.	Process Control (PROC_CTRL).....	58
6.2.7.2.	Login Session Management (SM_LOGIN).....	59
6.2.7.3.	RBAC Session Management (SM_RBAC) .....	60
6.2.8.	Resource Utilization .....	60
6.2.8.1.	Process Priority (RU_PRIO).....	60
6.2.8.2.	Maximum Quota (RU_QUOTA).....	61
6.2.9.	Protection of TOE Security Functions.....	61
6.2.9.1.	Protection Functions (PROT_FUNCS) .....	61
6.3.	TOE Summary Specification Rationale .....	61
APPENDIX A: References .....		69
APPENDIX B: Acronyms.....		70

## Table of Tables

<b>TABLE 5.1 – AUDITABLE EVENTS</b>	18
<b>TABLE 5.2.1 – SECURITY ASSURANCE REQUIREMENTS</b>	37
<b>TABLE 6.2.1.2.1 – AUDIT EVENT TYPES AND SYSTEM CALLS</b>	44
<b>TABLE 6.2.1.2.2 – SELF-AUDITING PROCESSES</b>	45
<b>TABLE 6.2.2.1.1 – FILE SYSTEM OBJECTS DAC ACCESS MODE PERMISSIONS</b>	48
<b>TABLE 6.2.2.2.1 – IPC OBJECTS DAC ACCESS MODE PERMISSIONS</b>	51
<b>TABLE 6.2.3.1.1 – FILE SYSTEM OBJECTS MAC ACCESS MODE PERMISSIONS</b>	52
<b>TABLE 6.2.3.2.1 – IPC OBJECTS MAC ACCESS MODE PERMISSIONS</b>	53
<b>TABLE 6.3.1 – SFR TO SF MAPPING RATIONALE</b>	61

# 1. ST Introduction

## 1.1. ST Reference

**Title:** HP-UX 11i v3 Security Target for CC evaluation against the COTS Compartmentalized Operations Protection Profile - Operating Systems [CCOPP-OS], Version 1.6, Hewlett-Packard Development Company, L.P., October 31, 2009.

**Keywords:** Protection Profile, Separation of Duties, Role Based Access, Discretionary Access, Mandatory Access, Compartmentalized Operations, Information Protection, General Purpose Operating System.

## 1.2. TOE Reference

**TOE:** HP-UX 11i v3 Update 3 Virtual Server Operating Environment (VSE-OE) Common Criteria Evaluated Configuration

## 1.3. TOE Overview

### 1.3.1. TOE Type

The TOE is Hewlett-Packard's implementation of UNIX-based operating system that executes on the entire range of PA-RISC based HP 9000 and Itanium-based HP Integrity servers in both stand-alone and networked environment in multi-user mode of operations. On cell-based HP 9000 and HP Integrity platforms, the TOE can execute with hard partition (nPartition) and logical partition (vPartition) configurations.

### 1.3.2. Major Security Features

The TOE provides the following major security features that are sufficient for controlling a community of authenticated users:

- User Identification and Authentication
- Discretionary Access Control (DAC)
- Mandatory Access Control (MAC)
- Role Based Access Control (RBAC)
- Security Audit
- Resource Utilization
- Object Reuse Protection
- Trusted Recovery
- Environment Constraints Based Access Control

The TOE provides protection against unsophisticated technical attacks such as an unauthorized access to the TOE by masquerading as another user. However, the TOE is not expected to totally protect against malicious abuse of authorized privileges or sophisticated technical attacks such as denial-of-service.

## **1.4. TOE Description**

### **1.4.1. TOE Physical Scope**

The TOE is HP-UX 11i v3 Virtual Server Operating Environment (VSE-OE). The TOE executes on the hardware configurations described below. The TOE is an operating system. It does not include any hardware. It also does not include virtual partition monitor software that is required for operation in vPartition.

#### **Platforms**

The TOE executes on any supported single 64-bit computer system from the family of HP 9000 Servers and HP Integrity Servers – hereafter referred to collectively as “server”.

#### **Hard Partition**

On a cell-based server, the TOE executes in any nPartition configured within the server. Cell-based servers may be configured as one single large system or as multiple smaller systems by configuring nPartitions. Each nPartition defines a subset of server hardware resources to be used as an independent system environment. An nPartition includes one or more cells assigned to it (with processors and memory) and all I/O chassis connected to those cells. All processors, memory, and I/O in an nPartition are used exclusively by the software running in the nPartition. Thus, each nPartition has its own system boot interface and each nPartition boots and reboots independently. Each nPartition provides both hardware and software isolation, so that hardware or software faults in one nPartition do not affect other nPartitions within the same server complex.

#### **Logical Partition**

Within a server or any nPartition of a cell-based server, the TOE executes in any vPartition. A server or an nPartition of a cell-based server may be configured as one single large system or as multiple smaller systems by configuring vPartitions. Each vPartition defines a subset of available hardware resources on an nPartition or server to be used as an independent system environment. A vPartition includes one or more CPU-cores, a contiguous physical memory range in the multiple of configured granular size and one or more Local Bus Adapters (LBA). All processors, memory, and LBAs in a vPartition are used exclusively by the software running in the vPartition. Thus, each vPartition has its own system boot interface and each vPartition boots and reboots independently. Each vPartition provides software isolation, so that software faults in one vPartition do not affect other vPartitions within the same server or nPartition.

#### **Network Environment**

The TOE may be connected to other servers via a local Ethernet network, each executing the same version of the TOE and under the same administrative control. The TOE may also be connected to other [CCOPP-OS] or [CAPP] conformant systems, such as PCs or workstations, under the same administrative control and on the same local network. No other processors may be connected to the TOE, either directly by hard wire connection by, for example, cluster of servers or indirectly by, for example, a Wide Area Network or telephone cable to provide remote computer or network services.

#### **Console Access**

The preceding bullet is not intended to preclude system console connections through the use of a private LAN connection to a Guardian Service Processor. System console connections may be through either a serial line or through a Guardian Service Processor connection. Refer to A.PEER and A.LOCATE assumptions in section 3.4.

### **1.4.2. TOE Logical Scope**

The TOE executes with the following configurations. Refer to the Evaluated Configuration Guide [ECG] for details.

#### **User Shells**

The TOE supports user interaction via any of the supported Shells (including the POSIX, Bourne, C, and Korn Shells).

#### **File System**

The TOE includes the HFS and VxFS File Systems, but excludes Online VxFS.

#### **Pluggable Authentication Module**

The TOE includes support of the Pluggable Authentication Modules (PAM) framework, with the default configuration for authentication consisting of traditional user identity and password. Although the PAM framework permits other authentication modules, such as authentication through NT domain servers, LDAP or DCE, to be used, these are not included in the evaluated configuration.

#### **Window System**

The TOE executes with CDE and X-Window disabled.

#### **Network Applications**

The TOE includes socket based network functions and the following secure network applications (other network applications and services, such as NFS and NIS, are excluded):

- a) scp(1)
- b) sftp(1)
- c) ssh(1)

### **1.4.3. Major Security Features**

The major security features provided by the TOE are:

#### **User Identification and Authentication**

All users of the TOE are authenticated and held accountable for their security related actions. Each user is uniquely identified by the TOE. The TOE records security related events and the user associated with the event. The authentication features are supported by constraints on user-generation of passwords and an encryption mechanism.

#### **Discretionary Access Control (DAC)**



The TOE enforces Discretionary Access Control (DAC) policies between active entities (subjects) and passive entities (objects) based on subject identity and allowed actions on the object. The TOE implements DAC policies through both the traditional UNIX 'owner', 'group', and 'other' access mode permissions and a more granular access control list (ACL) mechanism controlled by the object's owner.

Except for kernel daemon processes that operate directly on behalf of the kernel, all subjects are associated with an authenticated user identity, and all named objects are associated with identity based protection attributes. These are used as the basis of DAC decisions, which control the access of subjects to objects.

### **Mandatory Access Control (MAC)**

The TOE enforces Mandatory Access Control (MAC) policies between active entities (subjects) and passive entities (objects) based on the compartment label of the subject and allowed actions on the object.

All subjects are associated with a compartment label. The subjects' compartment labels and the 'label' access restriction rules for the objects defined in a compartment access rule database are used as the basis for the MAC decisions, which control the access of subjects to the objects.

### **Role Based Access Control (RBAC)**

The TOE implements Role Based Access Control (RBAC) which breaks up the traditional one system administrator ('superuser') into a number of roles. The users may be assigned role(s). Each role is associated with zero or more authorizations for a privileged operation on an object. For example, a network administrator has a role that permits configuring network cards.

### **Security Audit**

The TOE provides mechanisms to record security relevant events. It allows detection of any attempts to bypass the protection mechanism. It acts as a deterrent against system abuses and exposing potential security breaches in the system. An authorized administrator may select the users and events for which audit record is collected from time to time.

### **Resource Utilization**

The TOE implements resource allocation policies for system resources as a measure of resistance to resource depletion. Maximum amount of memory and disk space per user or subject can be set by an authorized administrator.

### **Object Reuse Protection**

An object reuse protection mechanism ensures that information is not inadvertently transferred between subjects when objects are re-allocated.

### **Trusted Recovery**

The TOE provides mechanisms for trusted recovery in the event of system failures or detected insecurities such as system database corruption.

### **Environmental Constrains Based Access Control**

The TOE can allow or deny access based upon environmental constraints such as time-of-day and port-of-entry

## 2. Conformance Claims

### 2.1. CC Conformance

This ST is CC version 3.1 Part 2 extended, and is Part 3 conformant with evaluation assurance level EAL4 augmented by Systematic Flaw Remediation (ALC\_FLR.3). It is Part 2 extended because [CCOPP-OS], which this ST is based upon, includes security functional requirements that are extensions to those found in CC version 3.1 Part 2 Security Functional Components.

### 2.2. PP Conformance

This ST is conformant with [CCOPP-OS] version 2.0. Conformance with [CCOPP-OS] is demonstrable with assurance level EAL4 as defined in CC Part 1.

### 2.3. PP Tailoring

The TOE security functional requirements are derived from [CCOPP-OS]. They have been tailored by performing operations required by [CCOPP-OS] as defined in Chapter 5. Assignments and selections performed by the Security Target are highlighted with **bold** fonts. Refinements performed by the Security Target are highlighted with ***bold italic*** fonts.

## 3. Security Problem Definition

### 3.1. Introduction

This chapter identifies the threats to the assets to be protected by the TOE and the operational environment, organizational security policies for which the TOE is appropriate, and the assumptions about the operational environment for the TOE.

This chapter this provides the basis for derivation of the security objectives described in chapter 4 and hence the specific security requirements listed in chapter 5.

### 3.2. Threats

#### 3.2.1. Assets

Assets requiring protection are fully conformant with [CCOPP-OS].

The TOE protects the information it stores and processes, its resources, and the services it provides to authorized users. The value of the assets merits moderately intensive penetration or masquerading attacks.

#### 3.2.2. Threat Agents

Threat agents are fully conformant with [CCOPP-OS].

Threat agents may be either authorized or unauthorized users of the TOE. Authorized users will vary in the degree of access rights and privileges they have been granted. In general, this security problem definition draws no distinction between different types of user; however, in certain specific instances the term *authorized administrator* is used to denote an individual who has been given responsibilities in respect of security administration of the TOE.

There are two broad categories of users with respect to these assumptions and threats:

The first category are persons who possess little technical skills, do not have access to sophisticated attack tools, have some rights of access, and are mostly trusted not to attempt to maliciously subvert the system nor maliciously exploit the information stored thereon. Users in this category may be motivated by curiosity to gain access to information for which they have no authorization.

The second category of users is technically skilled or has access to sophisticated attack tools and some may attempt to bypass system controls as a technical challenge or as a result of curiosity. CCOPP-OS conformant TOEs would generally be used in environments where these users are highly trusted not to attempt to maliciously subvert the system or to maliciously exploit the information stored thereon, or are restricted from gaining access by environmental measures.

#### 3.2.3. Threats addressed by the TOE

Threats addressed by the TOE are fully conformant with [CCOPP-OS].

**T.ACCESS** An authenticated user may gain unauthorized, non-malicious access to the TOE or a resource or to information directly controlled by the TOE via user error, system error, or an unsophisticated, technical attack.

**T.CRASH** The secure state of the TOE may be compromised in the event of a system crash, leading to corruption or loss of assets.

**T.DENIAL** The TOE may be subjected to an unsophisticated, denial-of-service attack.

**T.ENTRY** An individual, other than an authenticated user, may gain unauthorized, malicious access to TOE-controlled processing resources or information, via an unsophisticated, technical attack

**T.RECORD-EVENT** Security relevant events controlled by the TOE may not be recorded, and hence malicious activity may not be detected.

**T.RESOURCES** The shared, internal TOE resources may become exhausted due to system error or non-malicious user actions.

**T.ROLE-SEPARATION** The development and assignment of user roles may be done in a manner that undermines security, for example assigning users conflicting roles with respect to separation of duties.

**T.TOE-CORRUPTED** The security state of the TOE, as a result of an unsophisticated technical attack, may be intentionally corrupted to enable future insecurities.

**T.TRACEABLE** Security relevant events controlled by the TOE may not be traceable to the user or system process/processes associated with the event.

### 3.2.4. Threats addressed by the Operational Environment

Threats addressed by the operational environment are fully conformant with [CCOPP-OS].

**T.E.ADMIN-ERROR** Authenticated users or external threat agents may, through accidental discovery or directed search, discover errors or omissions and inadequacies in the security administration of the TOE, or other IT, which permit them to gain unauthorized access.

**T.E.DENIAL-SOPHISTICATED** The system may be subjected to a sophisticated, denial-of-service attack.

**T.E.ENTRY-NON-TECHNICAL** An individual, other than an authenticated user, may gain access to processing resources or information using non-technical means.

**T.E.ENTRY-SOPHISTICATED** An individual, other than an authenticated user, may gain access to processing resources or information using a sophisticated, technical attack.

**T.E.INSTALL** The system may be delivered or installed in a manner that undermines security.

**T.E.MALWARE** The confidentiality, integrity or availability of assets may be compromised as a result of the execution of malware (e.g. viruses, worms, Trojans, and so on).

### 3.3. Organizational Security Policies

The organizational security policies are fully conformant with [CCOPP-OS].

**P.ACCESS** Access rights by individual users to specific data objects are to be determined by the designated owner of the object, as laid down by the organization's security policy. These are to be based on the security attributes assigned to both the object and the individual user attempting access, as well as any environmental conditions that must also apply.

**P.ACCOUNTABILITY** The users are to be held accountable for their security-relevant actions.

**P.AUTHORIZED-USERS** Only those users who have been authorized to access the information within the system are to be able to access the system.

**P.COMPARTMENT** Access by users to information is restricted, based on the compartment label of the individual and the label-based access restrictions of the information. The access rules enforced prevents individuals from accessing information to which they are not authorized, in accordance with established information flow control policies.

**P.NEED-TO-KNOW** Access to information, and the ability to modify or destroy that information, is to be limited to those authorized individuals who have a "need to know" for that information.

**P.TRAINING** Authorized users are to be adequately trained, enabling them to (1) effectively implement organizational security policies with respect to their discretionary actions and (2) support the need for non-discretionary controls implemented to enforce these policies.

**P.USAGE** The organization's IT resources are used for only for authorized purposes.

### 3.4. Assumptions

The assumptions on the operating environment are fully conformant with [CCOPP-OS].

#### 3.4.1. Usage Aspects

**A.COMPARTMENT** Procedures exist for establishing the compartment label based restrictions of all information imported into or stored in the system.

**A.PEER** Any other systems with which the TOE communicates are assumed to be under the same management control and operate under the same security policy constraints.

### 3.4.2. Physical Aspects

**A.LOCATE** The processing resources of the TOE and connections to peripheral devices will be located within controlled access facilities which will prevent unauthorized physical access. It is also assumed that physical controls in place would alert the system authorities to the physical presence of attackers within the controlled space.

**A.PROTECT** The TOE hardware and software critical to security policy enforcement will be protected from unauthorized physical modification. Internal communication paths to access points such as terminals are assumed to be adequately protected.

### 3.4.3. Personnel Aspects

**A.ACCESS** Rights for users to gain access and perform operations on information are based on their membership in one or more roles. These roles are granted to the users by the RBAC Administrator. These roles accurately reflect the users' job function, responsibilities, qualifications, and/or competencies within the enterprise.

**A.COOP** Authorized users possess the necessary authorization to access at least some of the information managed by the TOE and are expected to act in a cooperating manner in a benign environment.

**A.MANAGE** There will be one or more competent individuals assigned to manage the TOE and the security of the information it contains.

**A.NO-EVIL-ADM** The system administrative personnel are not careless, wilfully negligent, or hostile, and will follow and abide by the instructions provided by the administrative documentation.

**A.USER-NEED** Authenticated users recognize the need for a secure IT environment.

**A.USER-TRUST** Authenticated users are generally trusted to perform discretionary actions in accordance with security policies.

## 4. Security Objectives

The security objectives are fully conformant with [CCOPP-OS].

### 4.1. Security Objectives for the TOE

**O.ACCOUNTABILITY** The TOE must ensure, for actions under its control or knowledge, that all TOE users can subsequently be held accountable for their security relevant actions.

**O.AUDITING** The TOE must record security relevant events in sufficient detail to help an administrator detect attempted security violations or potential mis-configuration of security functions that would leave IT assets at risk of compromise. The TOE must present this information to authorized administrators, and ensure that its confidentiality and integrity are protected.

**O.AVAILABLE** The TOE must protect itself from unsophisticated, denial-of-service attacks. This will include a combination of protection and detection.

**O.BYPASS** The TOE must prevent errant or non-malicious, authorized software or users from bypassing or circumventing TOE security policy enforcement.

**O.DETECT** The TOE must enable the detection of TOE specific insecurities.

**O.DISCRETIONARY-ACCESS** The TOE must control access to resources/objects based on identity of users. The TOE must allow authorized users to specify which resources may be accessed by which users.

**O.DUTY** The TOE must provide the capability of enforcing 'separation of duties', so that no single user has to be granted the right to perform all operations on important information.

**O.ENFORCEMENT** The TOE must be designed and implemented in a manner which ensures that the organizational policies are enforced in the target environment.

**O.ENTRY** The TOE must prevent logical entry to the TOE by persons or processes without authority for such access, using unsophisticated technical methods.

**O.HIERARCHICAL** The TOE must allow hierarchical definitions of roles to facilitate administration of the TOE, i.e., the ability to define roles in terms of other roles.

**O.MANAGE** The TOE must provide all the functions and facilities necessary to support the authorized administrators, ensuring that only authorized administrators can access such functionality.

**O.MANDATORY-ACCESS** The TOE must control access to resources based upon the compartment based access restriction rules for the information being accessed and the compartment of the subject.

**O.RECOVER** The TOE must provide for recovery to a secure state following a system failure, discontinuity of service, or detection of insecurity.

**O.RESOURCES** The TOE must protect itself from user or system errors that result in shared resource exhaustion.

**O.RESIDUAL-INFORMATION** The TOE must ensure that any information contained in a protected resource is never revealed when the resource is reused by a different subject.

**O.ROLE** The TOE must prevent users from gaining access to and performing operations on its resources/objects unless they have been granted access by the resource/object owner or they have been assigned to a role (by an authorized administrator) which permits those operations.

## 4.2. Security Objectives for the Operational Environment

**O.E.AUDIT-MANAGE** Those responsible for the TOE must ensure that audit data is analyzed on a regular basis, and that audit logs are backed up and retained for subsequent analysis where needed, and that appropriate measures are taken to guard against loss of audit data.

**O.E.AUTHENTICATION** The IT environment must provide one or more additional authentication mechanisms that can be used by the TOE when making authentication decisions.

**O.E.CONNECT** Those responsible for the TOE must ensure that no connections to outside systems or users undermine the security of the assets it is intended to protect.

**O.E.CREDEN** Those responsible for the TOE must ensure that all access credentials, such as passwords or other authentication information, are protected by the users in a manner which maintains the security of the assets it is intended to protect.

**O.E.DENIAL-SOPHISTICATED** The TOE environment must maintain system availability in the face of sophisticated denial-of-service attacks, with a focus on detection and response.

**O.E.DETECT-SOPHISTICATED** The TOE environment must provide the ability to detect sophisticated attacks and the results of such attacks (e.g., corrupted system state).

**O.E.ENTRY-NON-TECHNICAL** The TOE environment must provide sufficient protection against non-technical attacks by other than authenticated users. The focus will be on prevention, with user awareness playing a major part.

**O.E.ENTRY-SOPHISTICATED** The TOE environment must sufficiently mitigate the threat of an individual (other than an authenticated user) gaining unauthorized access via sophisticated, technical attack, with a focus on detection and response.



**O.E.INSTALL** Those responsible for the TOE must ensure that the TOE is delivered, installed, managed, and operated in a manner which maintains the security of the assets it is intended to protect.

**O.E.MALWARE** Those responsible for the TOE must ensure that the risk of introduction of malware is mitigated by the deployment of appropriate countermeasures (IT and procedural).

**O.E.PHYSICAL** Those responsible for the TOE must ensure that those parts of the TOE critical to security policy are protected from physical attack which might compromise IT security.

**O.E.SECURITY-ATTRIBUTES** Those responsible for the TOE must establish and implement procedures to ensure that security attributes (e.g. subject compartment labels, access rights, roles) are correctly determined and applied.

**O.E.TRUSTED-ADMIN** Those responsible for the TOE must implement procedures to ensure that adequate trust is established in the TOE administrators, that they are made aware of their responsibilities for security, and that they are given appropriate training so as to effectively discharge those responsibilities.

**O.E.USER-AWARENESS** Those responsible for the TOE must implement procedures to ensure that adequate trust is established in the TOE users, that they are made aware of their responsibilities for security, and that they are given appropriate training so as to effectively discharge those responsibilities.

### **4.3. Security Objectives Rationale**

The security objectives rationale presented in [CCOPP-OS] Section 7.1 satisfies the security objectives rationale requirement for this ST.

## 5. Security Requirements

The following sections provide the definitions of the security functional requirements for the TOE drawn from [CCOPP-OS]. Operations performed on functional components are highlighted as described in Section 2.3 PP Tailoring.

### 5.1. Security Functional Requirements

#### 5.1.1. Audit (FAU)

##### 5.1.1.1. Audit Data Generation (FAU\_GEN.1)

**FAU\_GEN.1.1** The TSF shall be able to generate an audit record of the auditable events listed in column “Auditable Events” of Table 5-1.

**Table 5.1 – Auditable Events**

<b>SFR</b>	<b>Auditable Events</b>
<b>FAU</b>	
FAU_GEN.1	Start-up and shutdown of the audit functions.
FAU_GEN.2	None
FAU_SAR.1	Reading of information from audit records.
FAU_SAR.2	Unsuccessful attempts to read information from audit records.
FAU_SAR.3	None
FAU_SEL.1	All modifications to the audit configuration that occur while the audit collection functions are operating.
FAU_STG.1	None
FAU_STG.3	Actions taken due to exceeding of a threshold.
FAU_STG.4	Actions taken due to the audit storage failure.
<b>FDP</b>	
FDP_ACC.1-A	None
FDP_ACF.1-A	All requests to perform an operation on an object covered by the DAC Policy.
FDP_ACC.1-B	None
FDP_ACF.1-B	All requests to perform an operation on an object covered by the RBAC Policy.
FDP_ETC.1	All attempts to export information.
FDP_IFC.1	None
FDP_IFF.1	All requests to perform an operation on an object covered by the MAC Policy.
FDP_ITC.1	All attempts to import user data.
FDP_RIP.2	None
FDP_RIP.CCOPP	None
<b>FIA</b>	
FIA_AFL.1	Actions taken when the threshold of unsuccessful authentication failures is reached.

<b>SFR</b>	<b>Auditable Events</b>
FIA_ATD.1	None
FIA_SOS.1	Rejection or acceptance by the TSF of any tested secret.
FIA_UAU.2	All use of the authentication mechanism.
FIA_UAU.CCOPP	None
FIA_UAU.6	All use of the authentication mechanism.
FIA_UAU.7	None
FIA_UID.2	All use of the authentication mechanism, including the identity provided during successful attempts.
FIA_USB.1	Success and failure of binding user security attributes to a subject.
<b>FMT</b>	
FMT_MSA.1-A	All modifications of the values of DAC security attributes.
FMT_MSA.1-B	All modifications of the values of RBAC security attributes.
FMT_MSA.1-C	All modifications of the values of MAC security attributes.
FMT_MSA.1-D	All modifications of the values of user security attributes.
FMT_MSA.2	All offered and rejected values for an RBAC security attribute.
FMT_MSA.3-A	Modifications of the default settings of permissive or restrictive rules. All modifications of the initial value of DAC security attribute.
FMT_MSA.3-B	Modifications of the default settings of permissive or restrictive rules. All modifications of the initial value of RBAC security attribute.
FMT_MSA.3-C	Modifications of the default settings of permissive or restrictive rules. All modifications of the initial value of MAC security attribute.
FMT_MTD.1-A	All modifications to the Audit Trail.
FMT_MTD.1-B	All modifications to the subset of audited events.
FMT_MTD.1-C	All initial assignments of authentication data.
FMT_MTD.1-D	All modifications to the values of the authentication data.
FMT_MTD.1-E	All modifications to TOE Access Banner.
FMT_MTD.1-F	All modifications to Role Definitions.
FMT_MTD.3	All rejected values of Role Definitions.
FMT_REV.1-A	All attempts to revoke user attributes.
FMT_REV.1-B	All attempts to revoke object attributes.
FMT_SAE.1	All attempts to change limits.
FMT_SMF.1	Use of management functions.
FMT_SMR.2	Every use of the rights of a role.
<b>FPT</b>	
FPT_FLS.1	Failure of the TSF.
FPT_ITC.CCOPP	None
FPT_ITI.CCOPP	None
FPT_RCV.1	The fact that a failure or service discontinuity occurred. Resumption of the regular operation. Type of failure or service discontinuity.

<b>SFR</b>	<b>Auditable Events</b>
FPT_RCV.4	If possible, the impossibility to return to a secure state after a failure of the TSF. If possible, the detection of a failure of a function.
FPT_STM.1	Changes to the time.
FPT_TEE.1	Execution of the tests of the underlying machine and the results of the tests.
FPT_TST.1	Execution of the TSF self tests and the results of the tests.
<b>FRU</b>	
FRU_PRS.1	Rejection of operation based on the use of priority within an allocation.
FRU_RSA.1	Rejection of allocation operation due to resource limits.
<b>FTA</b>	
FTA_LSA.1	All attempts at selecting a session security attribute.
FTA_MCS.1	Rejection of a new session based on the limitation of multiple concurrent sessions.
FTA_SSL.4	Termination of an interactive session by the user.
FTA_TAB.1	None
FTA_TAH.1	None
FTA_TSE.1	All attempts at establishment of a user session.

**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;
- b) The role that made possible the invocation of action;
- c) The compartment labels of subjects; and
- d) The additional information specified in the “Auditable Events” column of Table 5-1.

**5.1.1.2. User Identity Generation (FAU\_GEN.2)**

**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.

**5.1.1.3. Audit Review (FAU\_SAR.1)**

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

**FAU\_SAR.1.2** The TSF shall provide the audit records in a manner suitable for the user to interpret the information.

**5.1.1.4. Restricted Audit Review (FAU\_SAR.2)**

**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.

#### 5.1.1.5. Selectable Audit Review (FAU\_SAR.3)

**FAU\_SAR.3.1** The TSF shall provide the ability to apply selection and ordering of audit data based on the following attributes:

- a) User Identity;
- b) Object identity and type of access (where applicable);
- c) Role that enabled invocation of action;
- d) Subject compartment label;
- e) Date and time of audit event;
- f) **Terminal port;**
- g) **Set of event types;**
- h) **Set of system calls;**
- i) **Successful events;** and
- j) **Failed events;**

#### 5.1.1.6. Selective Audit (FAU\_SEL.1)

**FAU\_SEL.1.1** The TSF shall be able to select the set of audited events from the set of all auditable events based on the following attributes:

- a) User Identity;
- b) Users belonging to a specified role and access types (e.g., delete, insert) on a particular object.
- c) Subject identity;
- d) Object identity;
- e) Host identity;
- f) Event type;
- g) Subject compartment label;
- h) **System Call;**
- i) **Successful events;** and
- j) **Failed events.**

#### 5.1.1.7. Protected Audit Trail Storage (FAU\_STG.1)

**FAU\_STG.1.1** The TSF shall protect the stored audit records in the audit trail from unauthorized deletion.

**FAU\_STG.1.2** The TSF shall be able to prevent unauthorized modifications to the audit records in the audit trail.

#### 5.1.1.8. Actions in Case of Possible Audit Data Loss (FAU\_STG.3)

**FAU\_STG.3.1** The TSF shall generate an alarm to the authorized administrator if the audit trail exceeds **an authorized administrator selectable percentage of the storage capacity.**

#### 5.1.1.9. Prevention of Audit Data Loss (FAU\_STG.4)

**FAU\_STG.4.1** The TSF shall prevent audited events, except those taken by the authorized administrator if the audit trail is full.

## 5.1.2. User Data Protection (FDP)

### 5.1.2.1. Discretionary Access Control Policy (FDP\_ACC.1-A)

**FDP\_ACC.1.1-A** The TSF shall enforce the Discretionary Access Control (DAC) Policy on:

- a) **All subjects** acting on behalf of users
- b) **File System Objects and IPC objects**
- c) All operations among subjects and objects covered by the DAC policy.

**Application Note:** Network Interface Objects are not covered by the DAC policy. They are covered by the Mandatory Access Control (MAC) policy.

### 5.1.2.2. Discretionary Access Control Policy Rules (FDP\_ACF.1-A)

**FDP\_ACF.1.1-A** The TSF shall enforce the Discretionary Access Control (DAC) Policy to objects based on the following:

- a) The user identity and group membership(s) associated with a subject; and
- b) The following access control attributes associated with an object:
  - i) **For File System Objects, the Access Mode Permissions and the ACL**
  - ii) **For IPC Objects, the Access Mode Permissions**

**FDP\_ACF.1.2-A** The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

- a) **For HFS File System Objects:**
  - i) **If the object is associated with an HFS ACL, the effective user identity and group membership(s) associated with a subject are checked against ACL entries in the following order until access is granted or the end is reached (and then access is denied by default):**
  - ii) **Access is granted or denied according to the permissions of matching ACL entries bitwise-OR'd together if there is a match with one or more specific user, or specific group ACL entry;**
  - iii) **Access is granted or denied according to the permissions of the matching ACL entry if there is a match with a specific user, no specific group ACL entry;**
  - iv) **Access is granted or denied according to the permissions of matching ACL entries bitwise-OR'd together if there is a match with one or more no specific user, specific group ACL entry;**
  - v) **Access is granted or denied according to the permissions of the default no specific user, no specific group ACL entry.**
  - vi) **Otherwise, the user identity and group membership(s) associated with a subject are checked against the Access Mode Permissions in the following order until access is granted or the end is reached:**
    - 1) **Access is granted or denied according to the permissions if there is a match with object's owner class of user;**

- 2) Access is granted or denied according to the permissions if there is a match with the object's group class of user;
  - 3) Access is granted or denied according to the permissions of the object's other class of user.
- b) For VxFS File System Objects (see aclv(5) for notations such as user::):
- i) The effective user identity and group membership(s) associated with a subject are checked against ACL entries in the following order until access is granted or the end is reached (and then access is denied by default):
    - 1) Access is granted or denied according to the permissions in the user: : entry if there is a match with the object's owner class of user;
    - 2) Access is granted or denied according to the permissions in the user: uid: entry bitwise-AND'd with the class: entry if there is a match with an additional user ACL entry.
    - 3) Access is granted or denied according to the permissions in the group: : entry if there is a match with the object's group class of user;
    - 4) Access is granted or denied according to the permissions in the group: gid: entry bitwise-AND'd with the class: entry if there is a match with an additional group ACL entry;
    - 5) Access is granted or denied according to the permissions in the other: entry.
- c) For System V IPC and POSIX IPC Objects:
- i) The user identity and group membership(s) associated with a subject are checked against the Access Mode Permissions in the following order until access is granted or the end is reached (and then access is denied by default):
    - 1) Access is granted or denied according to the permissions if there is a match with the object's owner or (System V only) creator class of user;
    - 2) Access is granted or denied according to the permissions if there is a match with the object's group or (System V only) creator group class of user;
    - 3) Access is granted or denied according to the permissions of the object's other class of user.

**FDP\_ACF.1.3-A** The TSF shall explicitly authorize access of subjects to objects based on the following additional rules:

- a) **The subjects with DAC policy override privileges shall have access to all objects, overriding the rules specified in FDP\_ACF.1.2-A.**

**Application Note:** A subject may possess the DAC policy override privileges through the Role Based Access Control (RBAC) mechanism.

**FDP\_ACF.1.4-A** The TSF shall explicitly deny access of subject to objects based on the following additional rules:

- a) **Access by subjects to objects allowed by the rules specified in FDP\_ACF.1.2-A and FDP\_ACF.1.3-A is denied if Mandatory Access Control Policy (MAC) Rules specified in FDP\_IFF.1 deny such access.**

### 5.1.2.3. Role-Based Access Control Policy (FDP\_ACC.1-B)

**FDP\_ACC.1.1-B** The TSF shall enforce the Role-based Access Control (RBAC) Policy on:

- a) **All subjects** acting on behalf of users
- b) **File System Objects and IPC objects**
- c) All operations among subjects and objects covered by the RBAC policy.

### 5.1.2.4. Role-Based Access Control Policy Rules (FDP\_ACF.1-B)

**FDP\_ACF.1.1-B** The TSF shall enforce the Role-Based Access Control (RBAC) Policy to objects based on the following:

- a) User identity and authorized roles for the user; and
- b) Subject identity and role(s) which can invoke the subject; and
- c) Object identity and operations permitted on the objects for different roles.

**FDP\_ACF.1.2-B** The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: The subject invoking the operation on an object is assigned to a role whose privilege set includes the operation on the object.

**Application Note:** The subject is assigned to a role only if the user associated with the subject belongs to the role.

**FDP\_ACF.1.3-B** The TSF shall explicitly authorize access of subjects to objects based on **no other rules than those specified in FDP\_ACF.1.2-B.**

**FDP\_ACF.1.4-B** The TSF shall explicitly deny access of subjects to objects based on **no other rules than those specified in FDP\_ACF.1.2-B and FDP\_ACF.1.3-B.**

### 5.1.2.5. Export of User Data (FDP\_ETC.1)

**FDP\_ETC.1.1** The TSF shall enforce the Mandatory Access Control (MAC) Policy when exporting user data, controlled under the MAC policy, outside of the TOE, by enforcing the following rules:

- a) **A subject is allowed to export a file object when the subject is allowed to have read access based on the MAC and DAC policies on the file object.**

**FDP\_ETC.1.2** The TSF shall export the user data without the user data's associated security attributes.

### 5.1.2.6. Mandatory Access Control Policy (FDP\_IFC.1)

**FDP\_IFC.1.1** The TSF shall enforce the Mandatory Access Control (MAC) Policy on:

- a) **All subjects acting on behalf of users**
- b) **File System Objects, IPC objects, and Network Interface Objects**



- c) **All operations among subjects and objects covered by the MAC policy.**

**5.1.2.7. Mandatory Access Control Policy Rules (FDP\_IFF.1)**

**FDP\_IFF.1.1** The TSF shall enforce the Mandatory Access Control (MAC) Policy based on the following types of subject and information security attributes:

- a) The subject compartment label; and
- b) The compartment label of the object containing the information.

**Application Note:** The compartment label of the File System Object is a conceptual label that is defined by the set of access restriction rules for each compartment in the TOE.

**FDP\_IFF.1.2** The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold:

- a) **If the subject compartment labels include the object compartment labels, information flow from the object to the subject is permitted.**
- b) **If the object compartment labels include the subject compartment labels, information flow from the subject to the object is permitted.**

**Application Note:** TOE configuration may allow subjects and objects to be associated with one compartment, where the requirements shall be read: a) If the subject compartment label is the same as (conceptual) object compartment label, information flow control from the object to the subject is permitted. b) If the object compartment label is the same as the subject compartment label, information flow from the subject to the object is permitted.

**FDP\_IFF.1.3** The TSF shall enforce **no additional rules.**

**FDP\_IFF.1.4** The TSF shall explicitly authorize an information flow based on the following rules:

- a) **The subjects with MAC policy override privileges shall have access to all objects, overriding the rules specified in FDP\_IFF.1.2.**
- b) **The subjects associated with the INIT compartment shall have access to all objects, overriding the rules specified in FDP\_IFF.1.2.**

**Application Note:** System administration tasks are usually performed by the subject associated with the INIT compartment where the information flow is permitted between the subject and all objects under the control of the MAC policy.

**Application Note:** A subject may possess the MAC policy override privileges through the Role Based Access Control (RBAC) mechanism.

**FDP\_IFF.1.5** The TSF shall explicitly deny an information flow based on the following **rules:**

- a) **Information flow between controlled subject and controlled information allowed by the rules specified in FDP\_IFF.1.2 and FDP\_IFF.1.4 is denied if Discretionary Access Control (DAC) Policy Rules specified in FDP\_ACF.1-A deny such access.**

#### **5.1.2.8. Import of User Data (FDP\_ITC.1)**

**FDP\_ITC.1.1** The TSF shall enforce the Mandatory Access Control (MAC) Policy when importing user data, controlled under the MAC policy, from outside of the TOE.

**FDP\_ITC.1.2** The TSF shall ignore the security attributes associated with the user data when imported from outside the TOE.

**Application Note:** Compartment label of the imported user data is determined based on the access restrictions rules defined in the TOE.

**FDP\_ITC.1.3** The TSF shall enforce the following rules when importing user data controlled under the MAC policy from outside the TOE:

- a) **A subject is allowed to import a file object into a directory when the subject is allowed to have write access based on the MAC and DAC policies on the directory.**

#### **5.1.2.9. Object Residual Information Protection (FDP\_RIP.2)**

**FDP\_RIP.2.1** The TSF shall ensure that any previous information content of a resource is made unavailable upon the allocation of the resource to all objects.

#### **5.1.2.10. Subject Residual Information Protection (FDP\_RIP.CCOPP)**

**FDP\_RIP.CCOPP.1** The TSF shall ensure that any previous information content of a resource is made unavailable upon the allocation of the resource to all subjects.

### **5.1.3. Identification and Authentication (FIA)**

#### **5.1.3.1. Authentication Failure Handling (FIA\_AFL.1)**

**FIA\_AFL.1.1** The TSF shall detect when **an authorized administrator configurable positive integer** within **1 - 999 consecutive** unsuccessful authentication attempts occur related to **login attempts**.

**FIA\_AFL.1.2** When the defined number of unsuccessful *consecutive* authentication attempts has been surpassed, the TSF shall **lock the user account**.

**Application Note:** FIA\_AFL.1.1 and FIA\_AFL.1.2 have been refined to handle *consecutive* authentication failures.

#### **5.1.3.2. User Attribute Definition (FIA\_ATD.1)**

**FIA\_ATD.1.1** The TSF shall maintain the following list of security attributes belonging to individual users:

- a) User Identifier;
- b) Group Memberships;
- c) Authentication Data;
- d) Compartment Labels;
- e) User Roles;
- f) Default Active Role Set;
- g) **Home directory;**
- h) **Login program;**
- i) **Audit flag;**
- j) **Audit tag; and**
- k) **Boot flag**

#### 5.1.3.3. Verification of Authentication Data (FIA\_SOS.1)

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

- a) For each attempt to use the authentication mechanism, the probability that a random attempt will succeed is less than one in 1,000,000;
- b) For multiple attempts to use the authentication mechanism during a one minute period, the probability that a random attempt during that minute will succeed is less than one in 100,000; and
- c) Any feedback given during an attempt to use the authentication mechanism will not reduce the probability below the above metrics.

#### 5.1.3.4. User Authentication Before Any Action (FIA\_UAU.2)

**FIA\_UAU.2.1** The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

#### 5.1.3.5. Support for Multiple Authentication Mechanisms (FIA\_UAU.CCOPP)

**FIA\_UAU.CCOPP.1** The TSF shall provide support for **the required use of the authentication mechanisms other than only passwords** to support user authentication *without transmission of user authentication data between the TOE and another trusted IT product*.

**Application Note:** This SFR has been refined for the capability of the TSF to integrate the additional authentication mechanism without transmission of user authentication data. The refinement is not intended to preclude the support for the additional authentication mechanisms that require transmission of user authentication data.

**FIA\_UAU.CCOPP.2** The TSF shall authenticate any user's claimed identity according to the *user authentication policies enforced by the additional authentication mechanisms configured by the authorized administrator*.

**Application Note:** This SFR has been refined to allow the authorized administrator to select the user authentication policies and the authentication mechanisms to enforce the least privilege policy.

#### **5.1.3.6. Re-Authentication (FIA\_UAU.6)**

**FIA\_UAU.6.1** The TSF shall re-authenticate the user under the conditions requests to change the authentication secrets, and the following additional conditions:

- a) **Requests to change the user identity**

#### **5.1.3.7. Protected Authentication Feedback (FIA\_UAU.7)**

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

#### **5.1.3.8. User Identification Before Any Action (FIA\_UID.2)**

**FIA\_UID.2.1** The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

#### **5.1.3.9. User-Subject Binding (FIA\_USB.1)**

**FIA\_USB.1.1** The TSF shall associate the following user security attributes with subjects acting on the behalf of that user:

- a) The user identity which is associated with auditable events;
- b) The user identity or identities which are used to enforce the Discretionary Access Control Policy;
- c) The group membership or memberships used to enforce the Discretionary Access Control Policy;
- d) The compartment labels used to enforce the Mandatory Access Control Policy;
- e) The roles used to enforce the Role-based Access Control Policy; and
- f) **The current working directory.**

**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) The subject user identity associated with auditable events is set to the corresponding user identity.
- b) The real and effective subject user identity or identities which are used to enforce the Discretionary Access Control Policy is set to the corresponding user identity or identities;
- c) The real and effective group identities used to enforce the Discretionary Access Control Policy are set to the user's primary group membership;
- d) The subject compartment labels are set to the user compartment labels;
- e) The subject active role set to enforce the Role-Based Access Control Policy is set to the user's default active role set.
- f) **The current working directory is set to the user's home directory.**

**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) Only authorized administrators shall be able to change the user identity and group memberships of a subject acting on his or her behalf to that of another valid user;
- b) ***Except where prohibited by restrictions on the corresponding mount point, such as 'nosuid' flag***, a subject's effective user identity is changed to the owner of a file executed with its set-user-identity permission bit enabled;
- c) ***Except where prohibited by restrictions on the corresponding mount point, such as 'nosuid' flag***, a subject's effective group identity is changed to the owning group of a file executed with its set-group-identity permission bit enabled;
- d) **Except when allowed by the compartment change privilege, a subject is restricted to change its compartment; and**
- e) **Except when allowed by the change subject identity privilege, a subject is restricted to change its Active Role Set;**
- f) **The subject user identity associated with auditable events shall remain unchanged during the course of the user session when the subject user identity used to enforce the Discretionary Access Control Policy changes.**

**Application Note:** At the time of the initial association of user security attributes with subjects, the subject user identity associated with auditable events and the subject user identity used to enforce the Discretionary Access Control Policy are identical. During the course of the user session, the subject may change its user identity used to enforce the Discretionary Access Control Policy. However the subject user identity associated with auditable events must remain unchanged to ensure that the individual user is held accountable for the security relevant actions.

#### **5.1.4. Security Management (FMT)**

##### **5.1.4.1. Management of DAC Object Security Attributes (FMT\_MSA.1-A)**

**FMT\_MSA.1.1-A** The TSF shall enforce the Discretionary Access Control (DAC) Policy to restrict the ability to modify the DAC attributes associated with a named object to:

- a) **A subject acting as the owner or creator of the object may modify the permissions in the Access Mode Permissions and the ACL entries;**
- b) **A subject acting as the owner or creator of the object (and, for a File System Object, at the same time having the change ownership privilege) may change the ownership of the object;**
- c) **A subject acting as an authorized administrator may change permissions and the ownership of the object.**

**Application Note:** MAC Policy does not apply to DAC object security attributes. DAC attributes are protected solely by DAC Policy and can be modified by a subject if DAC Policy permits such access to the object even if MAC Policy permits no access by the subject to the object.

##### **5.1.4.2. Management of RBAC Object Security Attributes (FMT\_MSA.1-B)**

**FMT\_MSA.1.1-B** The TSF shall enforce the Role Based Access Control (RBAC) Policy to restrict the ability to modify the RBAC attributes associated with a named object to the object owner and **authorized administrators**.

#### 5.1.4.3. Management of Object Compartment Labels (FMT\_MSA.1-C)

**FMT\_MSA.1.1-C** The TSF shall enforce the Mandatory Access Control (MAC) Policy to restrict the ability to modify the compartment label-based access restriction rules associated with an object to the **authorized administrators**.

#### 5.1.4.4. Management of User Security Attributes (FMT\_MSA.1-D)

**FMT\_MSA.1.1-D** The TSF shall enforce the Discretionary, Role-Based and Mandatory Access Control Policies to restrict the ability to initialize and modify the user security attributes, other than authentication data, to authorized administrators.

#### 5.1.4.5. Secure RBAC Security Attributes (FMT\_MSA.2)

**FMT\_MSA.2.1** The TSF shall ensure that only secure values are accepted for RBAC security attributes.

**Application Note:** This component is referenced as FMT\_MSA.2 without the component number.

#### 5.1.4.6. DAC Static Attribute Initialization (FMT\_MSA.3-A)

**FMT\_MSA.3.1-A** The TSF shall enforce the Discretionary Access Control (DAC) Policy to provide restrictive default values for security attributes that are used to enforce the Discretionary Access Control Policy.

**FMT\_MSA.3.2-A** The TSF shall allow the **authorized administrators and the owner or creator of an object** to specify alternative initial values to override the default values when an object or information is created.

**Application Note:** Among all the object types under control of DAC Policy, only System V IPC Object is associated with creator (and owner) identity.

#### 5.1.4.7. RBAC Static Attribute Initialization (FMT\_MSA.3-B)

**FMT\_MSA.3.1-B** The TSF shall enforce the Role-Based Access Control (RBAC) Policy to provide restrictive default values for security attributes that are used to enforce the Role-Based Access Control Policy.

**FMT\_MSA.3.2-B** The TSF shall allow the **authorized administrators** to specify alternative initial values to override the default values when an object or information is created.

#### 5.1.4.8. MAC Static Attribute Initialization (FMT\_MSA.3-C)

**FMT\_MSA.3.1-C** The TSF shall enforce the Mandatory Access Control (MAC) Policy to provide restrictive default values for security attributes that are used to enforce the Mandatory Access Control Policy.

**FMT\_MSA.3.2-C** The TSF shall allow the **authorized administrators** to specify alternative initial values to override the default values when an object or information is created.

**5.1.4.9. Management of Audit Trail (FMT\_MTD.1-A)**

**FMT\_MTD.1.1-A** The TSF shall restrict the ability to create, delete, and clear the audit trail, *and set the limit of the audit trail that triggers generation of the alarm* to authorized administrators.

**Application Note:** This SFR has been refined to include the management function to set the adjustable limit of the audit trail that triggers generation of the alarm specified in FAU\_STG.3.1.

**5.1.4.10. Management of Audited Events (FMT\_MTD.1-B)**

**FMT\_MTD.1.1-B** The TSF shall restrict the ability to modify or observe the set of audited events to authorized administrators.

**5.1.4.11. Management of Authentication Data – Initialization (FMT\_MTD.1-C)**

**FMT\_MTD.1.1-C** The TSF shall restrict the ability to initialize the authentication data to authorized administrators.

**5.1.4.12. Management of Authentication Data – Modification (FMT\_MTD.1-D)**

**FMT\_MTD.1.1-D** The TSF shall restrict the ability to modify the authentication data to the following:

- a) authorized administrators; and
- b) users authorized to modify their own authentication data

**5.1.4.13. Management of TOE Access Banner (FMT\_MTD.1-E)**

**FMT\_MTD.1.1-E** The TSF shall restrict the ability to modify the TOE Access Banner to authorized administrators.

**5.1.4.14. Management of Role Definitions (FMT\_MTD.1-F)**

**FMT\_MTD.1.1-F** The TSF shall restrict the ability to create and modify the Role Definitions, Role Attributes, Role Hierarchies, and Constraints among Role Relationships to authorized administrators.

**5.1.4.15. Secure Role Definition Values (FMT\_MTD.3)**

**FMT\_MTD.3.1** The TSF shall ensure that only secure values are accepted for Role Definitions, Role Attributes, Role Hierarchies and Constraints among Role Relationships.

#### 5.1.4.16. Revocation of User Attributes (FMT\_REV.1-A)

**FMT\_REV.1.1-A** The TSF shall restrict the ability to revoke all security attributes associated with the users under the control of TSF to authorized administrators.

**FMT\_REV.1.2-A** The TSF shall enforce the rules:

- a) The immediate revocation of security-relevant authorizations; and
- b) **The revocation of security-relevant authorizations by removing or modifying user security attributes (e.g., user name) and by changing the user's password, which is effective from the next time the user attempts authentication.**

**Application Note:** The immediate revocation of security-relevant authorizations is achieved by removing or modifying the user security attributes and/or changing the user's password and then forcing the user to log off.

#### 5.1.4.17. Revocation of Object Security Attributes (FMT\_REV.1-B)

**FMT\_REV.1.1-B** The TSF shall restrict the ability to revoke all security attributes associated with objects under the control of the TSF to users authorized to modify the security attributes by the Discretionary, Role-Based or Mandatory Access Control policies.

**FMT\_REV.1.2-B** The TSF shall enforce the rules:

- a) The access rights associated with an object shall be enforced when an access check is made;
- b) The rules of the Mandatory Access Control policy are enforced on all future operations.

#### 5.1.4.18. Time-Limited Authorization (FMT\_SAE.1)

**FMT\_SAE.1.1** The TSF shall restrict the capability to specify an expiration time for user account and authenticators to the authorized administrator.

**Application Note:** Authorized administrator specifiable expiration time for these attributes enforces the time-limited authorization on users to perform any authenticated action. A user with the expired account will not be able to interact with any part of the TSF, except login attempts.

**FMT\_SAE.1.2** For each of these security attributes, TSF shall be able to:

- a) For user account – disable account and require administrator action to re-enable
- b) For authenticators – require owner of authenticator to establish a new value before proceeding with authenticated action

after the expiration time for the indicated security attribute has passed.

**Application Note:** The capability of the TSF to disable a user account and to require authorized administrator to re-enable the disabled account ensures that no active user account can exist without explicit administrative action. Requiring the owner of authenticator to establish a new value after the expiration reduces the chance of account fraud.



#### **5.1.4.19. Specification of Management Functions (FMT\_SMF.1)**

**FMT\_SMF.1.1** The TSF shall be capable of performing the following security management functions:

- a) Management of Object Security Attributes;
- b) Management of User Security Attributes;
- c) Management of Authentication Data;
- d) Management of Audit Trail;
- e) Management of Audited Events;
- f) Management of TOE Access Banner; and
- g) Management of Role Definitions, including Role Hierarchies and constraints.

#### **5.1.4.20. Security Roles (FMT\_SMR.2)**

**FMT\_SMR.2.1** The TSF shall maintain the roles:

- a) authorized administrator;
- b) object owner;
- c) users authorized by the Discretionary Access Control Policy to modify object security attributes;
- d) users authorized by the Mandatory Access Control Policy to modify object security attributes; and
- e) users authorized to modify their own authentication data.

**FMT\_SMR.2.2** The TSF shall be able to associate users with roles.

**FMT\_SMR.2.3** The TSF shall ensure that the following conditions for (a) Roles of Object Owners and (b) the set of RBAC administrative roles are satisfied.

- a) Object Owners can modify security attributes for only the objects they own;
- b) The set of RBAC administrative roles can modify security attributes for all objects under the control of TOE (since they automatically inherit the privileges of all Object Owners)

#### **5.1.5. Protection of the TOE Security Functions (FPT)**

##### **5.1.5.1. Failure with Preservation of Secure State (FPT\_FLS.1)**

**FPT\_FLS.1.1** The TSF shall preserve a secure state when the following types of failures occur:

- a) The entire RBAC database containing data on privileges assigned to a role, users authorized for a role, role constraints and relationships, or some specific tables containing subsets of these data are off-line, corrupt or inaccessible;

##### **5.1.5.2. Subset Inter-TSF Confidentiality During Transmission (FPT\_ITC.CCOPP)**

**FPT\_ITC.CCOPP.1** The TSF shall support the protection of authentication information transmitted from the TSF to another trusted IT product from unauthorized disclosure during transmission.

### 5.1.5.3. Subset Inter-TSF Detection of Modification (FPT\_ITI.CCOPP)

**FPT\_ITI.CCOPP.1** The TSF shall support the capability to verify the integrity of authentication information transmitted between the TSF and another trusted IT product and perform **retransmission of authentication information** if modifications are detected.

### 5.1.5.4. Manual Recovery (FPT\_RCV.1)

**FPT\_RCV.1.1** After **the following TSF failures**, the TSF shall enter a maintenance mode where the ability to return the TOE to a secure state is provided.

- a) The database containing the privilege data is not on-line or the particular data table is inaccessible or corrupt.
- b) The database containing the role membership information is not on-line or the particular data table is inaccessible or corrupt.

### 5.1.5.5. Function Recovery (FPT\_RCV.4)

**FPT\_RCV.4.1** The TSF shall ensure that the following functions and failure scenarios have the property that the function either completes successfully, or for the indicated failure scenarios, recovers to a consistent and secure state:

- a) For the function that checks whether a specified privilege is assigned to any role: a failure scenario where the database containing the privilege data is not on-line or the particular data table is inaccessible or corrupt.
- b) For the function that checks whether a specified role has been assigned to a particular user: a failure scenario where the database containing the role membership information is not on-line or the particular data table is inaccessible or corrupt.

### 5.1.5.6. Reliable Time Stamps (FPT\_STM.1)

**FPT\_STM.1.1** The TSF shall be able to provide reliable time stamps.

**Application Note:** The TSF maintains time stamps to a granularity of one second. This granularity, combined with the inherent ordering of audit trails, has proven sufficient to provide meaningful time stamps in audit records.

### 5.1.5.7. Testing of External Entities (FPT\_TEE.1)

**FPT\_TEE.1.1** The TSF shall run a suite of tests periodically during normal operation *and* at the request of an authorized administrator to check the fulfillment of the security assumptions provided by the abstract machine that underlies the TSF.

**FPT\_TEE.1.2** If the test fails, the TSF shall **generate an alarm to the authorized administrator**.

#### 5.1.5.8. TSF Testing (FPT\_TST.1)

**FPT\_TST.1.1** The TSF shall run a suite of self tests periodically during normal operation, at the request of the *authorized administrator*, and when invocation of access rights on **File System Objects and IPC Objects through RBAC mechanism** occurs to demonstrate the correct operation of the TSF.

**FPT\_TST.1.2** The TSF shall provide *authorized administrators* with the capability to verify the integrity of TSF data.

**FPT\_TST.1.3** The TSF shall provide *authorized administrators* with the capability to verify the integrity of stored TSF executable code.

**Application Note:** These SFRs have been refined to identify the role (authorized administrator) that is allowed to execute the self-tests.

#### 5.1.6. Resource Utilization (FRU)

##### 5.1.6.1. Limited Priority of Service (FRU\_PRS.1)

**FRU\_PRS.1.1** The TSF shall assign a priority to each subject in the TSF.

**FRU\_PRS.1.2** The TSF shall ensure that each access to **processing capability** shall be mediated on the basis of the subject's assigned priority.

##### 5.1.6.2. Maximum Quotas (FRU\_RSA.1)

**FRU\_RSA.1.1** The TSF shall enforce maximum quotas of the following resources: **memory and disk space** that **an individual user and a subject** can use **simultaneously**.

**Application Note:** Enforcing maximum quota for disk space and memory reduces the risk of resource depletion.

#### 5.1.7. TOE Access (FTA)

##### 5.1.7.1. Limitation on Scope of Selectable Attributes (FTA\_LSA.1)

**FTA\_LSA.1.1** The TSF shall restrict the scope of these session security attributes:

- a) User role based on **active role set defined for the user**; and
- b) Compartment label based on **the point of entry (console terminal/network interface through which the user has gained access to the TOE)**

**Application Note:** This ensures that the user session security attributes used to enforce Role Based Access Control (user role) and Mandatory Access Control (compartment label) are restricted based on the configuration defined by the authorized administrator.

**Application Note:** Root user can switch to any user without standard identification and authentication mechanism and can create sessions for the user with the compartment label which

the user would not be assigned when the user gains access to the TOE through regular login process.

#### **5.1.7.2. Basic Limitation on Multiple Concurrent Sessions (FTA\_MCS.1)**

**FTA\_MCS.1.1** The TSF shall restrict the maximum number of concurrent sessions that belong to the same user.

**FTA\_MCS.1.2** The TSF shall enforce, by default, a limit of **authorized administrator selected maximum number of** sessions per user.

**Application Note:** The maximum number of sessions per user is configurable by an authorized administrator.

#### **5.1.7.3. User-Initiated Termination (FTA\_SSL.4)**

**FTA\_SSL.4.1** The TSF shall allow user-initiated user-termination of the user's own interactive session.

#### **5.1.7.4. Default TOE Access Banners (FTA\_TAB.1)**

**FTA\_TAB.1.1** Before establishing a user session, the TSF shall display an advisory warning message regarding unauthorized use of the TOE.

#### **5.1.7.5. TOE Access History (FTA\_TAH.1)**

**FTA\_TAH.1.1** Upon successful session establishment, the TSF shall display the **date, time, and location** of the last successful session establishment to the user.

**FTA\_TAH.1.2** Upon successful session establishment, the TSF shall display the **date, time, and location** of the last unsuccessful attempt to session establishment and the number of unsuccessful attempts since the last successful session establishment.

**FTA\_TAH.1.3** The TSF shall not erase the access history information from the user interface without giving the user an opportunity to review the information.

#### **5.1.7.6. TOE Session Establishment (FTA\_TSE.1)**

**FTA\_TSE.1.1** The TSF shall be able to deny session establishment based on the following:

- a) Default active role set for the user being empty
- b) Session establishment outside the authorized administrator selected time periods (time of day and day of week)

**Application Note:** In the context of the requirement (a), the term session shall be interpreted as one within a regular login session during which one or more authorized roles are activated for a user. Establishment of a session (during a regular login session) that requires authorized roles shall be denied if a user has no authorized roles.

## 5.2. Security Assurance Requirements

The evaluation assurance level for the TOE is EAL4, augmented by Systematic Flaw Remediation ALC\_FLR.3, as summarized in Table 5.2.1 – Security Assurance Requirements.

**Table 5.2.1 – Security Assurance Requirements**

Assurance Class	Assurance Components
ADV: Development	ADV_ARC.1 Security Architecture Description
	ADV_FSP.4 Complete Functional Specification
	ADV_IMP.1 Implementation Representation of the TSF
	ADV_TDS.3 Basic Modular Design
AGD: Guidance Documents	AGD_OPE.1 Operational User Guidance
	AGD_PRE.1 Preparative Procedures
ALC: Life-Cycle Support (Augmented with ALC_FLR.3)	ALC_CMC.4 Production Support, Acceptance Procedures and Automation
	ALC_CMS.4 Problem Tracking CM Coverage
	ALC_DEL.1 Delivery Procedures
	ALC_DVS.1 Identification of Security Measures
	ALC_FLR.3 Systematic Flaw Remediation
	ALC_LCD.1 Developer Defined Life-Cycle Model
	ALC_TAT.1 Well-Defined Development Tools
ASE: Security Target Evaluation	ASE_CCL.1 Conformance Claims
	ASE_ECD.1 Extended Components Definition
	ASE_INT.1 ST Introduction
	ASE_OBJ.2 Security Objectives
	ASE_REQ.2 Derived Security Requirements
	ASE_SPD.1 Security Problem Definition
	ASE_TSS.1 TOE Summary Specification
ATE: Tests	ATE_COV.2 Analysis of Coverage
	ATE_DPT.2 Testing: Security Enforcing Modules
	ATE_FUN.1 Functional Testing
	ATE_IND.2 Independent Testing - Sample
AVA: Vulnerability Assessment	AVA_VAN.3 Focused Vulnerability Analysis

## 5.3. Security Requirements Rationale

The Security Requirements Rationale presented in [CCOPP-OS] Section 7.2 satisfies the security functional requirements rationale for this ST. In addition to the EAL4 security assurance components required by [CCOPP-OS], it was determined that augmentation by Systematic Flaw Remediation (ALC\_FLR.3) is appropriate for the TOE evaluation.

## 6. TOE Summary Specifications

### 6.1. Introduction

The following TOE summary specification charts will track the Security Function (SF) and Security Functional Requirement (SFR) to provide a clear and consistent high-level definition of the TOE Security Functions and Assurance Measures.

#### 6.1.1. Concepts and Terminology

##### 6.1.1.1. Subjects, Sessions and Privileges

A subject in the TOE is an active entity, generally in the form of a user process, which causes information to flow amongst objects.

A process has a number of security relevant attributes, which are used by the TOE to control a user's access to the TOE (via sessions) and to enforce the TOE's security policies. The security relevant attributes of a process include:

- a) the process ID
- b) the parent process ID
- c) the process group ID
- d) the process' real and effective user IDs
- e) The process' real and effective group IDs
- f) a group access list
- g) the process' compartment label
- h) the process' permitted, effective and retained fine-grained privilege sets
- i) an audit tag (dynamically assigned at session creation)
- j) the current working directory

A user gains initial access to the TOE via login at a terminal that involves authentication of the user. A successful login results in the creation of a login user session, which consists of a group of one or more processes.

The first process created in a session is known as a session leader (or process group leader), and its process group ID is set equal to its process ID. All other processes in the same session share the same process group ID. A parent process ID of a process is the process ID of its parent process.

The other security relevant attributes (such as process' real and effective user IDs and the current working directory) of the session leader process are set to those associated with the user authenticated during login, that is:

- a) the real and effective user IDs are set equal to the user's user ID.
- b) the real and effective group IDs are set equal to the user's group ID.
- c) the group access list is set equal to the set of supplementary group IDs.
- d) the compartment label is set to the compartment label associated with the user.
- e) the permitted, effective and retained fine-grained privilege sets is set to the *basic fine-grained privilege set*.
- f) the audit tag is dynamically generated and assigned.

- g) the current working directory is set equal to the user's home directory.

All security relevant attributes of a process (except the process, parent process and process group IDs) are inherited from the parent process.

After login, further sessions may be created by the user (e.g., background jobs), some of which may outlive the lifetime of the initial login session. All further session leader processes will inherit the above security relevant attributes that are associated with their parent process.

Whenever a process executes an executable object, the effective user ID, group IDs compartment label and fine-grained privilege sets may be changed. However, the audit tag will not be changed, thus maintaining user accountability for actions.

It may be allowed for a user to switch from one session to another session, which is associated with a different user ID. This will require full authentication of the new user ID. However, the audit tag will not be changed, thus maintaining the initial user's accountability for actions.

In order to perform certain security critical actions, typically those that affect other users, a user must either be a root user or have authorized administration status by means of assigned Roles. The appropriate fine-grained privileges must be associated in the effective fine-grained privilege set of the process that is performing the action on behalf of the user.

A process may possess the *PRIV\_CHOWN* fine-grained privilege, which means that the process can change the ownership of files that are currently owned by the user associated with the effective user ID of the process.

A process may possess the *PRIV\_CHGCMPT* security relevant system capability, which means that the process can change its compartment label.

#### **6.1.1.2. Objects**

An object is a passive container or receiver of information that may be categorized as one of several object types. Access to an object potentially implies access to the information contained within the object.

The TOE implements access control mechanisms for the following types of named objects:

- a) File System Objects, as follows:
  - i) regular (or ordinary) files
  - ii) (device) special files (character and block)
  - iii) directories
  - iv) named pipes
  - v) symbolic links
- b) System V IPC (Inter-Process Communication) and POSIX IPC objects, as follows:
  - i) message queues
  - ii) shared memory
  - iii) semaphores
- c) Network Interface Objects

Subsequent reference to objects in this document is restricted to the named objects listed in the previous paragraph.

Every object has an owning user and an owning group. The owning user is initially the user who created the object and the owning group is typically a default group associated with the owning user.

### 6.1.1.3. Fine-Grained Privileges

The TOE implements fine-grained privilege model to split the power of root users into a set of fine-grained privileges. Each fine-grained privilege grants a process that possesses that privilege the right to a certain set of restricted services provided by the kernel. Fine-grained privileges can be managed internally by a process with *privilege bracketing*. *Privilege bracketing* is the practice of enabling, or *raising*, a privilege only while the privilege is needed, then disabling, or *lowering*, the privilege. The privilege that a process has raised determine which sensitive system call services the process can invoke.

*Basic privileges* are granted by default to all processes. *Root replacement privileges* are the privileges that provide the powers associated with a process that has effective user ID of zero. *Root replacement* privileges are granted by default to any process with an effective user ID of zero. *Policy override privileges* override MAC rules. *Policy configuration privileges* control how the fine-grained privileges are configured.

Each process has three privilege sets associated with it:

*Permitted Privilege Set* (p) is the maximum set of privileges that a process can raise. The process can remove any privilege from this set, but cannot add a privilege to this set. The privileges from this set can be added to the *effective privilege set* of the process. This set is also referred to as *potential privilege set*.

*Effective Privilege Set* (e) is the set of privileges that are currently active for the process. A process can modify this set to keep only the necessary privileges in this set at any given time. Any privilege in this set can be removed, but only privileges in the process's *permitted privilege set* can be added. This set is always a subset of the *permitted privilege set* of the process.

*Retained Privilege Set* (r) is the set of privileges retained when a process calls *execve(2)*. The process can remove any privilege from this set, but can't add any privilege to this set. This set is always a subset of the *permitted privilege set* of the process.

An executable object in the TOE that depends on the use of privilege is registered as a privilege-aware program with *privilege\_start flag* (pf), *Minimum Permitted* (pm), *Maximum Permitted* (PM), *Minimum Retained* (rm) and *Maximum Retained* (RM) *Privilege Sets*. These sets are used to compute the Permitted (p'), Retained (r') and Effective (e') Privilege sets of the process that execs an privilege-aware executable object are as follows:

$$p' = (r \cup pm) \cap PM$$

$$r' = (r \cup rm) \cap RM \cap p'$$

If pf is set  $e' = p'$  else  $e' = \text{empty-set}$

where symbols  $\cup$  and  $\cap$  denote union and intersection respectively.

Privilege sets of the process that execs a non privilege-aware executable objects are as follows:



$$p' = r' = e' = r$$

#### 6.1.1.4. Discretionary Access Control (DAC) Mechanism

The TOE implements two standard access control mechanisms, which control discretionary access between subjects and objects according to access permissions, as follows:

- a) the traditional UNIX access mode permission mechanism, which applies to all named object
- b) an Access Control List (ACL) mechanism which, for File System Objects only, further qualifies the access given by the access mode permissions.

There are two types of ACL mechanism implementations, one for HFS File Systems and one for VxFS File Systems.

The TOE implements a discretionary access control (DAC) policy, whereby subjects associated with authenticated users gain access to objects in accordance with access permissions specified by the object owners or users with appropriate privileges.

The intent of the DAC policy is:

- a) to allow user to manage access control to its own objects.
- b) to protect user activities from undesired interference from other users.

#### 6.1.1.5. Mandatory Access Control (MAC) Mechanism

The TOE implements mandatory access control mechanism to control access between subjects and objects based on the compartments labels associated with the subjects.

The intent of the MAC policy is to restrict users' access to objects as defined by the system wide policy described in the compartment rule database.

File System objects and POSIX Shared Memory Objects do not have a compartment label associated with it. Instead, the rules that govern the access to these objects are associated with the name of the object. By default, these objects are accessible by a subject in any compartment. However, specific compartment configuration can define rules to restrict access to various file system objects.

System V and POSIX IPC objects except POSIX Shared Memory object are assigned the compartment label of the subject at the time of its creation. By default, subjects in a compartment cannot access these objects in another compartment unless explicitly configured otherwise.

The following is a list of primary Compartment components:

- **getrules(1M) command** to display MAC rules.
- **setrules(1M) command** to parse and activate the MAC rules.
- **MAC database files** are listed below

- /etc/cmpt/ – contains the files ending with **.rules** that are used to create compartment configurations.
- /etc/cmpt-rules.bin – contains the machine readable compartment rules
- /etc/cmpt-db – contains the mapping from compartment name to the ID

#### 6.1.1.6. Role Based Access Control (RBAC) Mechanism

TOE provides a Role Based Access Control (RBAC) mechanism to manage users, subjects, objects, and operations. RBAC groups users with common authorization needs into roles. Rather than assigning authorization directly to the user, the RBAC mechanism assigns authorizations to roles. As users are added to the system, they are assigned a set of roles which determine the actions they may perform and the resource they may access.

The following is a list of primary RBAC components:

- **privrun(1M) wrapper command** to run existing legacy applications without modifications and with varying privileges based on user authorizations
- **privedit(1M) command** to allow authorized users to edit files that are under access control
- **Access Control Policy Switch (ACPS)** to determine whether a subject is authorized to perform an operation on an object
- **Access Control Policy Module** to evaluate RBAC databases and apply mapping policies to service access control requests
- **Management Commands** to edit and validate RBAC database files, including:
  - a) **roleadm(1M)** - edits role information in RBAC database files
  - b) **authadm(1M)** - edits authorization information in RBAC database files
  - c) **cmdprivadm(1M)** - edits command authorizations and privileges in the privrun database
  - d) **rbacdbchk(1M)** - verifies syntax of RBAC and privrun database files
- **RBAC database files** are listed below
  - a) /etc/rbac/roles – contains the roles defined in RBAC
  - b) /etc/rbac/auth – contains the authorizations defined in RBAC
  - c) /etc/rbac/user\_role – contains the role assignments to users
  - d) /etc/rbac/role\_auth – contains the authorizations assigned to roles
  - e) /etc/rbac/cmd\_priv – contains the privileges/authorizations assigned to commands

The executive component of the TOE's RBAC mechanism is the **privrun(1M)** command, which is used to invoke existing administrative commands, applications, and scripts by means of a RBAC session within the current user session. The **privrun(1M)** command uses the Access Control Policy Switch to make access control requests based on a configuration file. An access request may be granted or denied based on a set of configuration files that define user-to-role and role-to-authorizations mappings.

If the access request is granted, **privrun(1M)** invokes the target command with additional privileges. These privileges – specifically, a new uid and or gid – are configured to allow the command to run successfully.

### **6.1.1.7. Initial and Secure States**

The initial state is achieved when the TOE is booted. This initial state has no subjects and is secure, since there are no object accesses in existence.

The initial state transitions to another state when the first user logs in thus creating a subject. This new state is also secure since the TOE implements boot authentication, whereby even root (or privileged) users accessing the TOE in single user state are authenticated.

All subsequent accesses, including all accesses in multi-user state, are mediated under the restrictions of the TOE's security policies, which preserve the secure state.

## **6.2. Security Functions**

### **6.2.1. Audit (AUD)**

#### **6.2.1.1. Audit Data Collection (AUD\_COLL)**

**AUD\_COLL.1** The TOE shall be capable of auditing all security relevant events that occur as a result of actions performed by the TOE on behalf of a user (system calls) on a per event and per user basis.

**AUD\_COLL.2** The TOE shall allow only an authorized administrator to turn the auditing capability on or off.

Note: Assumes that auditing is on when the TOE is operated in multi-user mode.

**AUD\_COLL.3** The TOE shall allow only an authorized administrator to turn the auditing capability on or off, on a per user basis, by setting the audit flag associated with the user to on or off, respectively.

**AUD\_COLL.4** The TOE shall protect the audit data so that it cannot be accessed by any user who is not authorized to do so.

**AUD\_COLL.5** The TOE shall log start-up and shut-down of the auditing functions.

**AUD\_COLL.6** The TOE shall audit records for the creation, assignment, modification, and deletion of roles, role authorizations, and command authorizations

**AUD\_COLL.7** The TOE shall audit records for the modification of MAC rule database.

#### **6.2.1.2. Audit Events (AUD\_EVENTS)**

**AUD\_EVENT.1** The TOE shall group system calls having a similar behavior into categories called 'event types' listed in Table 6.2.1.2.1 Audit Event Types and System Calls.

**AUD\_EVENT.2** The TOE shall allow only an authorized administrator to set or observe the auditing status of event types (on a per event type basis) and system calls (on a per system call basis) to one of the following:

- a) audit for success only
- b) audit for failure only

- c) audit for both success and failure
- d) do not audit

**AUD\_EVENT.3** The TOE's initial default selection of audit shall audit the success and failure of the following event types:

- a) admin
- b) logon
- c) moddac

**AUD\_EVENT.4** The TOE shall provide the capability for authorized administrators to create trusted applications so that auditing of system calls may be suspended or resumed at appropriate points in the process (known as a self-auditing process) and alternative or additional audit events are produced.

**AUD\_EVENT.5** The process listed in Table 6.2.1.2.2 Self-Auditing Processes shall be self-auditing.

**AUD\_EVENT.6** The TOE shall provide the following three event types, for use by an authorized administrator defined self-auditing processes, for which the auditing status may be set as specified in AUD\_EVENT.2:

- a) uevent1
- b) uevent2
- c) uevent3

**Table 6.2.1.2.1 – Audi Event Types and System Calls**

Event Type	Description of Action	Associated System Calls
admin	Log all administrative and privileged events	acct(2), adjtime(2), audctl(2), audswitch(2), audtag(2), clock_settime(2), _cnx_gsched_ctl(2), _cnx_p2p_ctl(2), getsym(2), kload(2), modadm(2), modload(2), moduload(2), modpath(2), modstat(2), mpctl(2), mem_res_grp(2), plock(2), privgrp(2), pset_assign(2), pset_bind(2), pset_setattr(2), reboot(2), sched_setparam(2), sched_setscheduler(2), serialize(2), setaudit(2), setaudproc(2), setdomainname(2), setevent(2), setprivgrp(2), setrlimit(2), setrlimit64(2), _set_mem_window(2), settimeofday(2), settune(2), spuctl(2), stime(2), swapon(2), toolbox(2), utssys(2)
close	Log all closings of objects	close(2), ksem_close(2), mq_close(2), munmap(2)
create	Log all creations of objects	creat(2), mkdir(2), mknod(2), msgget(2), pipe(2), pset_create(2), semget(2), shmat(2), shmget(2), symlink(2)
delete	Log all deletions of objects	ksem_unlink(2), mq_unlink(2), msgctl(2), pset_destroy(2), rmdir(2), semctl(2), shm_unlink(2)
ipcclose	Log all ipc close events	fdetach(2), shutdown(2)
ipccreat	Log all ipc create events	bind(2), socket(2), socket2(2), socketpair(2), socketpair2(2)

Event Type	Description of Action	Associated System Calls
ipcopen	Log all ipc open events	accept(2), connect(2), fattach(2)
login	Log all logins and logouts	logins and logouts
modaccess	Log all access modifications other than DAC	chdir(2), chroot(2), fchdir(2), link(2), lockf(2), lockf64(2), ptrace64(2), rename(2), sendfile(2), sendfile64(2), setcontext(2), setgid(2), setgroups(2), setpgid(2), setpgrp(2), setpgrp2(2), setpgrp3(2), setregid(2), setresgid(2), setresuid(2), setsid(2), setuid(2), shmctl(2), shmdt(2), ttrace(2), ulimit(2), unlink(2)
moddac	Log all modifications of object's DAC	acl(2), chmod(2), chown(2), fchmod(2), fchown(2), fsetacl(2), lchmod(2), lchown(2), putmsg(2), semop(2), semtimedop(2), setacl(2), umask(2)
open	Log all openings of objects	execv(2), execve(2), ftruncate(2), ftruncate64(2), ksem_open(2), mmap(2), mmap64(2), mq_open(2), open(2), ptrace(2), shm_open(2), truncate(2), truncate64(2)
process	Log all operations on processes	exit(2), fork(2), kill(2), mlock(2), mlockall(2), munlock(2), munlockall(2), nsp_init(2), rtprio(2), setpriority(2), sigqueue(2), vfork(2)
readdac	Log all DAC information reading	access(2), fstat(2), fstat64(2), getaccess(2), lstat(2), lstat64(2), stat(2), stat64(2)
removable	Log all removable media events (mounting and unmounting events)	exportfs(2), mount(2), umount(2), umount2(2), vfstmount(2)
uevent1 uevent2 uevent3	Log user defined events	See AUD_EVENTS.6

**Table 6.2.1.2.2 – Self-Auditing Processes**

Sel-Auditing Process	Description
audevent(1M)	Select events to be audited
audisp(1M)	Display the audit data
audsys(1M)	Start or halt the auditing system
authadm(1M)	Administers authorization information in RBAC databases
chfn(1)	Change finger entry
chsh(1)	Change login shell
cmdprivadm(1M)	Administers command/authorization/privilege mapping information in RBAC databases
fbackup(1M)	Selectively back up files
login(1)	The login utility
newgrp(1)	Change effective group
passwd(1)	Change password
privedit(1M)	Allows authorized users to edit files that are under access control.

Sel-Auditing Process	Description
privrun(1M)	Executive component of RBAC. Execute a legacy process after performing appropriate authorization check
roleadm(1M)	Administers role-related information in RBAC databases
setrules(1M)	Compartment Rules related information in MAC database
useradd(1M)	Add new user login account
userdel(1M)	Delete user login account
usermod(1M)	Modify user login account

### 6.2.1.3. Audit Records (AUD\_RECS)

**AUD\_RECS.1** The first time an audit event occurs in a process after an audit log is selected for use, the TOE shall write a process ID identification record into the audit log file which shall contain the following information:

- a) parent process ID
- b) audit tag
- c) real user ID
- d) real group ID
- e) effective user ID
- f) effective user group ID
- g) effective privileges
- h) permitted privileges
- i) retained privileges
- j) subject compartment label
- k) device name

**AUD\_RECS.2** For each event audited, the TOE shall record in the selected audit log file the following information:

- a) the system date and time that the audited event completes
- b) the event type
- c) the process ID of the process that causes the event
- d) the success or failure of the event
- e) event specific information, if required, as specified in AUD\_RECS.4 and AUD\_RECS.5

**AUD\_RECS.3** The date and time inserted into audit records shall be reliable.

**AUD\_RECS.4** For events generated by system calls, the event specific information which is recorded in the audit log file shall be 'the identity of the object' for all attempts to access FSO and IPC objects.

**AUD\_RECS.5** For events generated by self-auditing processes, the event specific information which is recorded in the audit log file shall be a high-level description of the event.

**AUD\_RECS.6** For each RBAC audit event type, the TOE shall record the following information:

- a) role assigned to user
- b) role authorization
- c) process authorization
- d) operation performed (process)
- e) object on which the operation was performed
- f) real user ID
- g) effective user ID
- h) subject compartment label

#### **6.2.1.4. Audit Logs Viewing (AUD\_VWNG)**

**AUD\_VWNG.1** The TOE shall provide the capability for only the authorized administrator to extract audit log data from a specified audit log file in accordance with one or more of the following selection criteria:

- a) a given user name
- b) a given terminal name
- c) a given set of event types
- d) a given set of system calls
- e) successful events
- f) failed events
- g) subject compartment label
- h) the event date and time at which to start the extraction of audit log data
- i) the event date and time at which to end the extraction of audit log data
- j) any combination of:
  - i) the role that enables the access
  - ii) object name associated with the event
  - iii) operation performed on the object

#### **6.2.1.5. Audit Log Files Maintenance (AUD\_MTNS)**

**AUD\_MTNS.1** The TOE shall collect audit records in:

- a) a *primary* log file, which is used initially by the TOE
- b) an optional *auxiliary* log file selected by an authorized administrator

**AUD\_MTNS.2** The TOE shall allow authorized administrator to specify the following parameters:

- a) Audit File Switch (AFS) size
- b) File Space Switch (FSS) size

**AUD\_MTNS.3** The TOE shall issue a warning on the console when the primary log file reaches a percentage, configurable by an authorized administrator, of the AFS size or the FSS size.

**AUD\_MTNS.4** When the AFS size or the FSS size is reached, the TOE shall attempt to switch to the auxiliary log file to collect audit records;

**AUD\_MTNS.5** If no auxiliary log file exists, the TOE shall periodically issue a warning on the console.

**AUD\_MTNS.6** When the space available on the file system(s) containing the primary log file and the auxiliary log file is exhausted, no audit records shall be collected. All auditable actions of unprivileged users shall be suspended. An authorized administrator shall be allowed to continue to carry out the operations without generating audit records.

**AUD\_MTNS.7** The TOE shall allow an authorized administrator to configure a percentage of total physical memory used for the temporary buffer of generated audit records before they are written to disk to minimize the amount of audit records that may be lost during a system crash.

**6.2.2. Discretionary Access Control (DAC)**

**6.2.2.1. Discretionary Access Control for File System Objects (DAC\_FS)**

**DAC\_FS.1** The TOE shall enforce discretionary access control (DAC) policy between subjects and File System objects by the access mode (owner/group/other) permissions and Access Control Lists (ACL) associated with each file system object.

**DAC\_FS.2** Each file system object is associated with the following attributes:

- a) An owning user identification (owner user ID)
- b) A group identification (group ID)
- c) A set of access permissions that specifies allowable access modes of the following three classes of (mutually independent) users:
  - i) The owner of the object identified by the owner user ID
  - ii) Any member of the group (except the owner) identified by the group ID
  - iii) Any other user (except the owner or any member of the group)
- d) Access Control Lists (ACL)

**DAC\_FS.3** The TOE shall allow selection of no access or any combination of the access mode permissions specified in the table below for access to a file system object independently for each class of user (owner/group/other).

**Table 6.2.2.1.1 – File System Objects DAC Access Mode Permissions**

File System Objects Access Mode Permissions		
Files	Directories	Special Files and Named Pipes
Read	Read	Read
Write	Write	Write
Execute	Search	-

**DAC\_FS.4** Whenever a process makes request to open a File System Object, the access mode permissions for that object shall be checked by the TOE, against the process’s effective user ID, effective group ID, and any group in the process’s group access list, to determine whether the process can access the object in the requested mode. Read, write and execute/search access to a File System Object is allowed by a process if any of the following is met, and no access is allowed if none of the conditions are met:

- a) The process’s effective user ID matches the object’s owner user ID and the appropriate access mode permission is set for the object’s owner class of user.



- b) The process's effective user ID does not match the object's owner user ID. The object group ID matches the process's effective group ID or a group in the process's group access list, and the appropriate access mode permission is set for the object's group class of user.
- c) The process's effective user ID does not match the object's user ID, the group in the process's group access list, and the appropriate access mode permission is set for the object's other class of user.
- d) The process has OWNER privilege.
- e) For read and execute/search access, the process has DACREAD privilege.
- f) For write access, the process has DACWRITE privilege.

**DAC\_FS.5** When a process creates a new File System Object, the object's owner user ID and the group user ID are set as follows:

- a) The owner user ID is set to the effective user ID of the process.
- b) The group ID is set to:
  - i) The group ID of the parent directory if the set-group-ID attribute is present in the parent directory's set of file protection attributes.
  - ii) The effective group ID of the process if the set-group-ID attribute is not present in the parent directory's set of the protection attributes.

**DAC\_FS.6** When a process creates a new File System Object, the set of access permissions which the process associates with the object are modified to remove any access permissions (limited to read, write, and execute) set in file mode creation mask (umask) of the process.

**DAC\_FS.7** A process shall be able to modify the access mode permissions associated with a File System Object, provided one or both of the following hold:

- a) The effective user ID of the process is equal to the file owner.
- b) The process has OWNER privilege.

**DAC\_FS.8** A process shall be able to change the user and group ownership of a File System Object, provided one or both of the following hold:

- a) The effective user ID of the process is equal to the file owner.
- b) The process has CHOWN privilege.

**DAC\_FS.9** Each VxFS ACL entry shall specify for one of owner, group, additional user ID, additional group ID, other or group class, and each HFS ACL entry shall specify for one user ID/group combination, a set of access permissions (as specified in the table for DAC\_FS.3) to the associated object, which may be zero or more of the following:

- a) Read
- b) Write
- c) Execute/Search

**DAC\_FS.10** Whenever an unprivileged process makes a request to open a File System Object, the ACL for that object shall be checked by the TOE to determine whether the process can access the object in the requested mode.

- a) For VxFS ACL, the TOE checks the ACL entries in the object's ACL against the process effective user ID and effective group ID respectively until a match is found, and grants or denies permissions accordingly, in the following order of precedence:
  - i) Permissions as specified in the user entry
  - ii) Permissions as specified in the additional user entry, bitwise-AND'd with those in the class entry
  - iii) Permissions as specified in the group entry
  - iv) Permissions as specified in the additional group entry, bitwise-AND's with those in the class entry
  - v) Permissions as specified in the other entry
- b) For HFS ACL, the TOE checks the ACL entries in the object's ACL against the process effective user ID and effective group ID and any group ID in the process's group access list, until a match is found for each effective user ID/group ID combination, and grants or denies permissions accordingly, in the following order of preference:
  - i) Specific user, specific group
  - ii) Specific user, no specific group
  - iii) No specific user, specific group
  - iv) No specific user, no specific group

**DAC\_FS.11** A process shall be able to modify the ACL associated with an object, provided one or both of the following hold:

- a) The process has ownership rights to the object
- b) The process has authorized administrator status

**DAC\_FS.12** When a process creates a new object, the TOE creates the base ACL entries to correspond with the object access mode permissions (as determined by DAC\_FS.6) as follows:

- a) For VxFS ACL:
  - i) Base ACL entry for the object's owner
  - ii) Base ACL entry for the object's group
  - iii) Base ACL entry for the object's group class
  - iv) Base ACL entry for the object's others
- b) For HFS ACL:
  - i) Base ACL entry for the object's owner
  - ii) Base ACL entry for the object's group
  - iii) Base ACL entry for the object's others

**DAC\_FS.13** When a process creates a new object, the TOE creates ACL entries corresponding with any default ACL entries of the directory in which the object is created.

**DAC\_FS.14** The TOE shall ensure that, irrespective of changes made by users to an object's access mode permissions or ACLs, the owner, group, others base ACLs for the object shall always correspond with the read, write and execute/search permissions set in the access mode permissions for the object's owner, group and others class of users.

**6.2.2.2. Discretionary Access Control for IPC Objects (DAC\_IPC)**

**DAC\_IPC.1** The TOE shall enforce discretionary access control (DAC) policy between subjects and IPC objects by the access mode (owner/group/other) permissions associated with IPC object.

**DAC\_IPC.2** Each IPC object is associated with the following attributes:

- a) An owning user identification (owner user ID)
- b) (System V only) A creator user identification (creator user ID)
- c) A group identification (group ID)
- d) (System V only) A creator group identification (creator group ID)
- e) A set of access permissions that specifies allowable access modes of the following three classes of (mutually independent) users:
  - i) The owner of the object identified by the object user ID
  - ii) (System V only) The creator of the object identified by the creator user ID
  - iii) Any member of the group (except the owner) identified by the group ID.
  - iv) (System V only) Any member of the creator group (except the owner) identified by the creator group ID.
  - v) Any other user (except the owner or any member of the group or (System V only) any member of the creator group)

**DAC\_IPC.3** The TOE shall allow selection of no access or any combination of the access mode permissions specified in the table below for access to an IPC object independently for each class of user (owner/group/other).

**Table 6.2.2.2.1 – IPC Objects DAC Access Mode Permissions**

System V and POSIX IPC Objects Access Mode Permissions		
Message Queue	Shared Memory	Semaphore
Receive	Attach for Read	Read
Send	Attach for Write	Alter

**DAC\_IPC.4** When an unprivileged process requests ‘receive(attach for read)/read’ and ‘send(attach for write)/alter’ access to a System V and POSIX IPC object, the access mode permissions for that object shall be checked by the TOE, against the process’s effective user ID, effective group ID, and any group in the process’s group access list, to determine whether the process can access the object in the requested mode. Receive/(attach for read) and send/(attach for write) access to System V and POSIX IPC objects is allowed by a process if any of the following conditions is met, and no access is allowed if none of the conditions are met:

- a) The process’s effective user ID matches the object’s owner user ID or (System V only) creator and the appropriate access mode permission is set for the object’s *owner* class of user.
- b) The process’s effective user ID does not match the object’s owner user ID or (System V only) creator user ID, the object group ID or (System V only) creator group ID matches the process’s effective group ID or a group in the process’s group access list, and the appropriate access mode permissions is set for the object’s *group* class of user.

- c) The process's effective user ID does not match the object's owner user ID or (System V only) creator user ID, the object group ID or (System V only) creator group ID does not match the process's effective group ID or a group in the process's group access list, and the appropriate access mode permission is set for the object's *other* class of user.
- d) The process has authorized administrator status.

**DAC\_IPC.5** When a process creates a new System V and POSIX IPC object, the object's owner user ID and (System V only) creator user ID and the object's group ID and (System V only) creator group ID shall be set as follows:

- a) The s owner user ID and (System V only) creator user ID shall be set to the effective user ID of the process.
- b) The group ID and (System V only) creator group ID shall be set to the effective group user ID of the process.

**DAC\_IPC.6** A process shall be able to modify the access mode permissions associated with a System V and POSIX IPC object, provided one or both of the following hold:

- a) The process has ownership rights, or (System V only) creator rights, or both ownerships and (System V only) creator rights to the object.
- b) The process is privileged, having authorized administrator status.

**DAC\_IPC.7** A process shall be able to change the user and group ownership of a System V or POSIX IPC object, provided one or both of the following fold:

- a) The process has ownership rights, or (System V only) creator rights, or both ownerships and (System V only) creator rights to the object.
- b) The process is privileged, having authorized administrator status.

### 6.2.3. Mandatory Access Control (MAC)

#### 6.2.3.1. Mandatory Access Control for File System Objects (MAC\_FS)

**MAC\_FS.1** TOE shall enforce Mandatory Access Control (MAC) policy between subjects and File System objects by the compartment label based access rules associated with each file system object

**MAC\_FS.2** Each file system object is associated with a set of access rules defined in compartment rules database for each compartment.

**MAC\_FS.3** The TOE shall allow selection of no access, all access or any combination of the access mode permissions specified in the table below for access to an object, for subjects belonging to one or more compartments.

**Table 6.2.3.1.1 – File System Objects MAC Access Mode Permissions**

File System Objects Access Mode Permissions		
Files	Directories	Special Files and Named Pipes
Read	Read	Read

Write	Write	Write
Create	Create	-
Unlink	Unlink	-
Execute	Search	-

**MAC\_FS.4** When a process requests to open a File System Object, the access mode permissions for that object shall be checked by the TOE, against the process compartment label to determine whether the process can access the object in the request mode. Read/Execute/Search, write, create and unlink access to a File System object is allowed by a process if any of the following conditions are met, and no access is allowed if none of the conditions are met:

- a) The process's compartment has a rule that allows the requested access to the object.
- b) The process's compartment has no rule that denies requested access to the object.
- c) The process's compartment label is INIT.
- d) The process has the privilege to override the MAC policy.

**MAC\_FS.5** When a process creates a new File System object, the compartment label based rules of the file system object is inherited from its parent directory.

**MAC\_FS.6** The TOE shall allow a process with appropriate privileges to modify the compartment label based access restriction rules associated with File System Objects.

**6.2.3.2. Mandatory Access Control for IPC Objects (MAC\_IPC)**

**MAC\_IPC.1** The TOE shall enforce Mandatory Access Control (MAC) policy between subjects and IPC objects by the compartment label based access rules associated with each IPC object.

**MAC\_IPC.2** Each IPC object is associated with a compartment label and a set of access rules defined in compartment rules database for each compartment.

**MAC\_IPC.3** The TOE shall allow selection of no access, all access or any combination of the access mode permissions specified in the table below for access to an object, for subjects belonging to one or more compartments.

**Table 6.2.3.2.1 – IPC Objects MAC Access Mode Permissions**

System V and POSIX IPC Objects		
Message Queue	Shared Memory	Semaphore
Receive	Attach for Read	Read
Send	Attach for Write	Alter

**MAC\_IPC.4** When a process requests access to a System V and POSIX IPC object, the access mode permissions for that object shall be checked by the TOE, against the process compartment label to determine whether the process can access the object in the requested mode. 'Receive/(attach for read)/read' and 'send/(attach for write)/alter' access to IPC objects is allowed

by a process if any of the following conditions are met, and no access is allowed if none of the conditions are met:

- a) The process' compartment has a rule that allows the requested access to the object.
- b) The process' compartment label is INIT.
- c) The process has the privilege to override the MAC policy.

**MAC\_IPC.5** When a process creates an IPC object, the compartment label of the object is set to the compartment label of the process's compartment label.

**MAC\_IPC.6** The TOE shall allow a process with appropriate privileges to change the compartment label of IPC objects.

### **6.2.3.3. Mandatory Access Control for Network Interface Objects (MAC\_NET)**

**MAC\_NET.1** The TOE shall enforce Mandatory Access Control (MAC) policy between subjects and Network Interface Objects by the compartment label based access rules associated with each object.

**MAC\_NET.2** Each Network Interface Object is associated with a compartment label and a set of access rules defined in compartment rules database for each compartment.

**MAC\_NET.3** The TOE shall allow selection of no access or all access to a Network Interface Object for subjects belonging to one or more compartments.

**MAC\_NET.4** When a process requests access to a Network Interface Object, the access is allowed by a process if any of the following conditions are met, and no access is allowed if none of the conditions are met:

- a) The process's compartment has a rule that allows the requested access to the object.
- b) The process's compartment label is INIT.
- c) The process has the privilege to override the MAC policy.

**MAC\_NET.5** The TOE shall allow a process with appropriate privileges to modify the compartment label of Network Interface Objects.

### **6.2.4. Role Based Access Control (RBAC)**

#### **6.2.4.1. Role Based Access Control (RBAC)**

**ACC\_RBAC.1** The TOE shall have the capability of administering roles, authorizations, and command's authorizations and privileges:

- a) Administration of roles:
  - i) Addition of roles to RBAC database;
  - ii) Deletion of roles from RBAC database;
  - iii) Modification of roles;
  - iv) Assignment of roles to users; and
  - v) Revocation of roles from users;

- b) Administration of authorizations:
  - i) Addition of authorizations to RBAC database;
  - ii) Deletion of authorizations from RBAC databases;
  - iii) Assignment of authorizations to roles; and
  - iv) Revocation of authorizations from roles;
- c) Administration of command's authorizations and privileges:
  - i) Addition of authorizations and privileges to commands; and
  - ii) Deletion of authorizations and privileges from commands;

**ACC\_RBAC.2** The TOE shall assume default values for the object security attributes if not specified when adding authorizations to and deleting authorizations from the RBAC database, and assigning authorizations to and revoking authorizations from roles.

**ACC\_RBAC.3** Assignment and revocation of roles and authorizations shall take effect immediately.

**ACC\_RBAC.4** The TOE shall enforce RBAC SFP through Access Control Policy (ACPS) subsystem by verifying user, role, and authorizations before allowing or denying the operation to take place on the object.

**ACC\_RBAC.5** The TOE shall define an (RBAC) administrator role to create, modify, and delete the following user security attributes.

- a) User Role Authorization
- b) Default Active Role Set

**ACC\_RBAC.6** The TOE shall restrict the ability to modify the following session security attribute to (RBAC) administrator role and session owner:

- a) Active Role set for a user

**ACC\_RBAC.7** The TOE shall use the following two step process to ensure that acceptable values are assigned to security attributes:

- a) Assignment of a role to a user; and
- b) Assignment of authorization to a role.

**ACC\_RBAC.8** The TOE shall permit only object owners and an authorized (RBAC) administrator to modify and/or revoke object security attributes.

**ACC\_RBAC.9** The TOE shall support Role hierarchies in the TOE RBAC mechanism.

**ACC\_RBAC.10** Assignment of security attributes to and revocation of security attributes from objects take effect immediately.

**ACC\_RBAC.11** The TOE shall define the set of (RBAC) authorized administrative roles as a role that is assigned the following authorizations:

- a) hpux.security.access.auth.\* for administration of the authorization information
- b) hpux.security.access.role.\* for administration of the role-related information
- c) hpux.security.access.privrun.\* for administration of the authorization and privilege information

## 6.2.5. Object Reuse

### 6.2.5.1. Object Reuse (OBJ\_REUSE)

**OBJ\_REUSE.1** The TOE shall ensure that that the disk blocks when allocated to the file-objects are scrubbed with zero.

**OBJ\_REUSE.2** The TOE shall ensure that that the memory blocks when allocated to the System V and POSIX IPC objects are scrubbed with zero.

**OBJ\_REUSE.3** The TOE shall ensure that the memory blocks when allocated to processes are scrubbed with zero.

## 6.2.6. Identification and Authentication (IA)

### 6.2.6.1. User Attributes (IA\_ATTR)

**IA\_ATTR.1** The TOE shall store the following identification and authentication attributes for each authorized user of the TOE:

- a) User name
- b) User ID
- c) Group ID
- d) Set of supplementary group IDs (optional)
- e) Compartment ID
- f) Audit flag
- g) Home directory
- h) Login program path name
- i) Boot flag
- j) Encrypted password
- k) Password minimum length
- l) Whether triviality check is performed on user-generated password
- m) Number of unsuccessful login attempts
- n) Maximum number of unsuccessful login attempts before the account is locked
- o) Account lock flag
- p) Account expiration time
- q) Password expiration time

**IA\_ATTR.2** The TOE shall store the identification and authentication attributes in a protected database. The access controls on the protected database shall be set such that only the authorized administrators can modify the identification and authentication attributes. Non-authorized users shall be able to modify their encrypted password entry through the trusted interface.

**IA\_ATTR.3** Any modification to IA\_ATTR.1 attributes (such as revocation of security attributes), except k), l), and q), will take place on the next login of the user. Modification to k)



and l) will take place the next time the password is changed. Modification to q) will take place after the next change of the password.

**IA\_ATTR.4** The TOE shall store the list of authorized roles in a protected database. The access controls on the protected database shall be set such that only the (RBAC) authorized administrator can modify the list of authorized roles.

**IA\_ATTR.5** The TOE shall have the capability to restrict the ability to create, modify, and delete the following list of TSF data to a set of (RBAC) administrative roles.

- a) User passwords
- b) Role definitions and role attributes

**IA\_ATTR.6** Any modification to IA\_ATTR.4 attributes (such as revocation of security attributes) will take place on the next RBAC session of the user.

**IA\_ATTR.7** The access controls on the protected database that keeps IA\_ATTR.5 shall be set such that only the authorized administrator can modify the list.

**IA\_ATTR.8** TOE shall allow only authorized administrators to see the user authentication information.

#### **6.2.6.2. User Authentication (IA\_AUTH)**

**IA\_AUTH.1** The TOE shall authenticate a user's identity before the user is permitted to gain access to the TOE's resources.

**IA\_AUTH.2** Successful authentication of a user shall require all of the following to be true:

- a) The user name entered by the user exists
- b) Except for the su(1) command executed by a previously authenticated superuser, in which case entry of a password is not required (see A.NO\_EVIL\_ADM), the password entered by the user, and one way encrypted by the TOE, is identical to the encrypted password stored by the TOE for the entered user name.
- c) Except for the root user account at the system console, the user account is not locked.

**IA\_AUTH.3** The TOE shall lock the user account if any of the following conditions are met:

- a) The user account has been explicitly locked by an authorized administrator.
- b) The user account has been inactive for a specified time interval (password expiration time in IA\_ATTR.1)
- c) The number of consecutive unsuccessful attempts to login to the user account exceeds the maximum allowed.

**IA\_AUTH.4** The TOE shall provide a boot authentication capability which shall require a user to enter a valid user name and password, for an account which has single-user login enabled, in order to boot the TOE into single-user mode.

**IA\_AUTH.5** The TOE shall provide the support for additional authentication mechanisms other than only passwords to authenticate any user's claimed identity without transmission of authentication data between the TSF and another trusted IT product.

**IA\_AUTH.6** The TOE shall re-authenticate the user when the user requests the change of the user identity.

**IA\_AUTH.7** The TOE shall allow only the authorized administrator to re-enable the account locked under conditions listed in IA\_AUTH.3.

**IA\_AUTH.8** The TOE shall require users to re-create passwords after the expiration before proceeding with authentication action.

### **6.2.6.3. User Identification (IA\_UID)**

**IA\_UID.1** The TOE shall uniquely identify a user by the user ID associated with that user's user name.

**IA\_UID.2** The TOE shall enforce individual accountability by associating the audit tag, associated with a user's user name, with all actions performed by the TOE on behalf of that user.

### **6.2.6.4. Password Selection, Generation, and Encryption (IA\_PW)**

**IA\_PW.1** The TOE shall allow users to create user-generated passwords.

Note: Only user-generated passwords are permitted in the evaluated configuration.

**IA\_PW.2** User-generated passwords shall comply with the following password construction criteria:

- a) Each password shall have at least six characters. Characters beyond the first eight are ignored.
- b) Each password shall contain at least two alphabetic characters and at least one numeric or special character.
- c) Each password shall differ from the user's user name, and any reverse or circular shift of that user name.
- d) New passwords shall differ from the old password by at least three characters.

**IA\_PW.3** The TOE shall one way encrypt passwords immediately after entry by a user.

**IA\_PW.4** The TOE shall neither display passwords in clear text during entry nor store user passwords in clear text.

## **6.2.7. Session Management (SM)**

### **6.2.7.1. Process Control (PROC\_CTRL)**

**PROC\_CTRL.1** Whenever a session leader process is created, the TOE shall ensure that the following attributes are inherited from the parent process:

- a) the real user ID
- b) the real group ID
- c) the effective user ID
- d) the effective group ID
- e) the group access list
- f) the process's current working directory
- g) the audit tag
- h) the compartment ID
- i) the active role set

**PROC\_CTRL.2** Whenever a session leader process is created, the TOE shall ensure that the process's attributes listed in ROC\_CTRL.1 are equal to those associated with the user authenticated during login, that is:

- a) Real and effective user IDs are set equal to the user's user ID.
- b) Real and effective group IDs are set equal to the user's group ID.
- c) Group access list is set to the set of supplementary group IDs.
- d) Audit tag is set equal to the audit tag of the parent process (process associated with the user authenticated during the login).
- e) Current working directory is set equal to the user's home directory.
- f) Current compartment ID is set based on the user's point of entry (console terminal/network interface through which the user has gained access to the TOE).
- g) Active role set is set based on the user ID.

**PROC\_CTRL.3** Whenever an executable object is executed by a process, the TOE shall ensure that:

- a) The new process effective user ID is set to the executable object's owner, if the set-user-ID access mode is associated with the executable object.
- b) The new process effective group ID is set to the executable object's group, if the set-group-ID access mode is associated with the executable object.

**Application Note:** Where prohibited by restrictions on the corresponding mount point, such as 'nosuid' flag, the effective user ID and the effective group ID of the new process remain unchanged.

**PROC\_CTRL.4** The TOE shall ensure that:

- a) Only an authorized administrator or privileged process shall be able to change the real and effective user/group IDs, the compartment label of a process, the active role set of a process, without re-authentication.
- b) The audit tag identity associated with the auditable events shall remain unchanged when the user ID used to enforce the Discretionary Access Control Policy changes.

### 6.2.7.2. Login Session Management (SM\_LOGIN)

**SM\_LOGIN.1** The TOE shall restrict the security attributes of a user session as follows

- a) Roles shall be based on user identity.
- b) Compartment shall be the default user compartment based on port of entry.

**SM\_LOGIN.2** The TOE shall enforce the successful start of a user session at an authorized administrator selected time periods (time of day and day of week).

**SM\_LOGIN.3** The TOE shall restrict the maximum number of concurrent sessions that belong to a user.

**SM\_LOGIN.4** The TOE shall enforce an authorized administrator selected maximum number of sessions per user.

**SM\_LOGIN.5** The TOE shall allow user-initiated termination of the interactive session.

**SM\_LOGIN.6** Before establishing a user session, the TOE shall display an advisory warning message regarding unauthorized use of the TOE. The advisory warning message is protected such that modification is allowed only by authorized administrators.

**SM\_LOGIN.7** Upon successful session establishment, the TOE shall display the date, time, and location of the last successful session establishment to the user.

**SM\_LOGIN.8** Upon successful session establishment, the TOE shall display the date, time, and location of the last unsuccessful attempt to session establishment and the number of unsuccessful attempts since the last successful session establishment.

**SM\_LOGIN.9** The TOE shall not erase the access history information from the user interface without giving the user an opportunity to review the information.

### **6.2.7.3. RBAC Session Management (SM\_RBAC)**

**SM\_RBAC.1** The TOE shall deny a RBAC session within a Login session based on the Default Active Role Set for the user being empty.

## **6.2.8. Resource Utilization**

### **6.2.8.1. Process Priority (RU\_PRIO)**

**RU\_PRIO.1** The TOE shall assign a relative priority to each process.

- a) The TOE shall allow a process to decrease its relative priority.
- b) The TOE shall allow a privileged process to increase its relative priority.
- c) The TOE shall allow a privileged process to change the relative priority of other process.

**RU\_PRIO.2** The TOE shall optionally assign a real time fixed priority to each process.

- a) The TOE shall allow a process to decrease its real time fixed priority.
- b) The TOE shall allow a privileged process to increase its real time fixed priority.
- c) The TOE shall allow a privileged process to change the real time fixed priority of other process.

**RU\_PRIO.3** When more than one processes request access to the processing capability simultaneously, the TOE shall grant the access to the highest priority process.

**6.2.8.2. Maximum Quota (RU\_QUOTA)**

**RU\_QUOTA.1** The TOE shall enforce maximum memory quota that a process can use.

**RU\_QUOTA.2** The TOE shall enforce maximum disk quota on a file-system that each user can use.

**RU\_QUOTA.3** The TOE shall enforce maximum disk quota on a file-system that each user group can use.

**RU\_QUOTA.4** The TOE shall enforce maximum memory quota for each IPC object type that each process can use.

**6.2.9. Protection of TOE Security Functions**

**6.2.9.1. Protection Functions (PROT\_FUNCS)**

**PROT\_FUNCS.1** The TOE shall allow an authorized administrator to run a test utility to confirm that:

- a) The TOE does not allow a user process to (1) read from or write to system vectors and unmapped areas of virtual memory and (2) write to read-only areas of virtual memory.
- b) The TOE maintains the integrity of executable code that implements the Security Functions.
- c) The TOE maintains the integrity of the user authentication and RBAC policy databases.

If the test fails, the TOE shall generate a warning message.

**PROT\_FUNCS.2** When one or more of the RBAC databases are off-line, corrupt, or inaccessible; all TOE commands, (and the associated security functions) that check for user roles and associated authorizations, shall fail gracefully. TOE shall allow only an authorized administrator to perform privileged operations in this state.

**Application Note:** RBAC-related TOE commands work on a per entry basis. As long as the relevant entry is valid, these commands work even if there is an invalid entry elsewhere in the RBAC database.

**6.3. TOE Summary Specification Rationale**

The following table shows the mapping between the Security Functions and Security Functional Requirements.

**Table 6.3.1 – SFR to SF Mapping Rationale**

SFR	Security Functions	Description
FAU		
FAU_GEN.1.1	AUD_COLL.1	Audit Data Collection

Common Criteria

SFR	Security Functions	Description
	AUD_COLL.5	Audit Data Collection
	AUD_COLL.6	Audit Data Collection
	AUD_COLL.7	Audit Data Collection
	AUD_EVENT.1	Audit Events
FAU_GEN.1.2	AUD_RECS.2	Audit Records
	AUD_RECS.4	Audit Records
	AUD_RECS.5	Audit Records
	AUD_RECS.6	Audit Records
FAU_GEN.2.1	AUD_RECS.1	Audit Records
	AUD_RECS.2	Audit Records
FAU_SAR.1.1	AUD_VWNG.1	Audit Logs Viewing
FAU_SAR.1.2	AUD_VWNG.1	Audit Logs Viewing
FAU_SAR.2.1	AUD_COLL.4	Audit Data Collection
	AUD_VWNG.1	Audit Logs Viewing
FAU_SAR.3.1	AUD_VWNG.1	Audit Logs Viewing
FAU_SEL.1.1	AUD_COLL.3	Audit Data Collection
	AUD_EVENT.1	Audit Events
	AUD_EVENT.2	Audit Events
	AUD_EVENT.3	Audit Events
	AUD_EVENT.4	Audit Events
	AUD_EVENT.5	Audit Events
	AUD_EVENT.6	Audit Events
FAU_STG.1.1	AUD_COLL.4	Audit Data Collection
	DAC_FS.1	Discretionary Access Control for File System Objects
	DAC_FS.2	Discretionary Access Control for File System Objects
	DAC_FS.3	Discretionary Access Control for File System Objects
	DAC_FS.4	Discretionary Access Control for File System Objects
	DAC_FS.9	Discretionary Access Control for File System Objects
	DAC_FS.10	Discretionary Access Control for File System Objects
FAU_STG.1.2	AUD_COLL.4	Audit Data Collection
	DAC_FS.1	Discretionary Access Control for File System Objects
	DAC_FS.2	Discretionary Access Control for File System Objects
	DAC_FS.3	Discretionary Access Control for File System Objects
	DAC_FS.4	Discretionary Access Control for File System Objects
	DAC_FS.9	Discretionary Access Control for File System Objects
	DAC_FS.10	Discretionary Access Control for File System Objects
FAU_STG.3.1	AUD_MTNS.1	Audit Log Files Maintenance
	AUD_MTNS.2	Audit Log Files Maintenance
	AUD_MTNS.3	Audit Log Files Maintenance
FAU_STG.4.1	AUD_MTNS.4	Audit Log Files Maintenance
	AUD_MTNS.5	Audit Log Files Maintenance
	AUD_MTNS.6	Audit Log Files Maintenance
<b>FDP</b>		

Common Criteria

SFR	Security Functions	Description
FDP_ACC.1.1-A	DAC_FS.1	Discretionary Access Control for File System Objects
	DAC_FS.14	Discretionary Access Control for File System Objects
	DAC_IPC.1	Discretionary Access Control for IPC Objects
FDP_ACF.1.1-A	DAC_FS.1	Discretionary Access Control for File System Objects
	DAC_FS.2	Discretionary Access Control for File System Objects
	DAC_FS.3	Discretionary Access Control for File System Objects
	DAC_FS.9	Discretionary Access Control for File System Objects
	DAC_IPC.1	Discretionary Access Control for IPC Objects
	DAC_IPC.2	Discretionary Access Control for IPC Objects
	DAC_IPC.3	Discretionary Access Control for IPC Objects
FDP_ACF.1.2-A	DAC_FS.4	Discretionary Access Control for File System Objects
	DAC_FS.10	Discretionary Access Control for File System Objects
	DAC_IPC.4	Discretionary Access Control for IPC Objects
FDP_ACF.1.3-A	DAC_FS.4	Discretionary Access Control for File System Objects
	DAC_IPC.4	Discretionary Access Control for IPC Objects
FDP_ACF.1.4-A	DAC_FS.4	Discretionary Access Control for File System Objects
	DAC_FS.10	Discretionary Access Control for File System Objects
	DAC_IPC.4	Discretionary Access Control for IPC Objects
	MAC_FS.2	Mandatory Access Control for File System Objects
	MAC_IPC.2	Mandatory Access Control for IPC Objects
FDP_ACC.1.1-B	ACC_RBAC.4	Role Based Access Control
FDP_ACF.1.1-B	ACC_RBAC.4	Role Based Access Control
FDP_ACF.1.2-B	ACC_RBAC.4	Role Based Access Control
FDP_ACF.1.3-B	ACC_RBAC.4	Role Based Access Control
FDP_ACF.1.4-B	ACC_RBAC.4	Role Based Access Control
FDP_ETC.1.1	MAC_FS.1	Mandatory Access Control for File System Objects
	MAC_FS.2	Mandatory Access Control for File System Objects
	MAC_FS.3	Mandatory Access Control for File System Objects
	MAC_FS.4	Mandatory Access Control for File System Objects
FDP_ETC.1.2	MAC_FS.2	Mandatory Access Control for File System Objects
FDP_IFC.1.1	MAC_FS.1	Mandatory Access Control for File System Objects
	MAC_IPC.1	Mandatory Access Control for IPC Objects
	MAC_NET.1	Mandatory Access Control for Network Interface Objects
FDP_IFF.1.1	MAC_FS.2	Mandatory Access Control for File System Objects
	MAC_IPC.2	Mandatory Access Control for IPC Objects
	MAC_NET.2	Mandatory Access Control for Network Interface Objects
FDP_IFF.1.2	MAC_FS.3	Mandatory Access Control for File System Objects
	MAC_FS.4	Mandatory Access Control for File System Objects
	MAC_IPC.3	Mandatory Access Control for IPC Objects
	MAC_IPC.4	Mandatory Access Control for IPC Objects
	MAC_NET.3	Mandatory Access Control for Network Interface Objects
	MAC_NET.4	Mandatory Access Control for Network Interface Objects

Common Criteria

SFR	Security Functions	Description
FDP_IFF.1.3	N/A	N/A
FDP_IFF.1.4	MAC_FS.4	Mandatory Access Control for File System Objects
	MAC_IPC.4	Mandatory Access Control for IPC Objects
	MAC_IPC.6	Mandatory Access Control for IPC Objects
	MAC_NET.4	Mandatory Access Control for Network Interface Objects
FDP_IFF.1.5	DAC_FS.4	Discretionary Access Control for File System Objects
	DAC_FS.10	Discretionary Access Control for File System Objects
	DAC_IPC.4	Discretionary Access Control for IPC Objects
FDP_ITC.1.1	MAC_FS.1	Mandatory Access Control for File System Objects
	MAC_FS.2	Mandatory Access Control for File System Objects
	MAC_FS.3	Mandatory Access Control for File System Objects
	MAC_FS.4	Mandatory Access Control for File System Objects
FDP_ITC.1.2	MAC_FS.2	Mandatory Access Control for File System Objects
FDP_ITC.1.3	MAC_FS.5	Mandatory Access Control for File System Objects
FDP_RIP.2.1	OBJ_REUSE.1	Object Reuse
	OBJ_REUSE.2	Object Reuse
	OBJ_REUSE.3	Object Reuse
FDP_RIP.CCOPP.1	OBJ_REUSE.1	Object Reuse
	OBJ_REUSE.2	Object Reuse
	OBJ_REUSE.3	Object Reuse
<b>FIA</b>		
FIA_AFL.1.1	IA_AUTH.3	User Authentication
FIA_AFL.1.2	IA_AUTH.3	User Authentication
FIA_ATD.1.1	IA_ATTR.1	User Attributes
	IA_ATTR.3	User Attributes
	ACC_RBAC.1	Role Based Access Control
FIA_SOS.1.1	IA_AUTH.3	User Authentication
	IA_PW.1	Password Selection, Generation, and Encryption
	IA_PW.2	Password Selection, Generation, and Encryption
FIA_UAU.2.1	IA_AUTH.1	User Authentication
	IA_AUTH.2	User Authentication
	IA_AUTH.3	User Authentication
	IA_AUTH.4	User Authentication
	IA_PW.3	Password Selection, Generation, and Encryption
FIA_UAU.CCOPP.1	IA_AUTH.5	User Authentication
FIA_UAU.CCOPP.2	IA_AUTH.5	User Authentication
FIA_UAU.6.1	IA_AUTH.6	User Authentication
FIA_UAU.7.1	IA_PW.4	Password Selection, Generation, and Encryption
FIA_UID.2.1	IA_AUTH.1	User Authentication
	IA_AUTH.2	User Authentication
	IA_AUTH.4	User Authentication
	IA_UID.1	User Identification



Common Criteria

SFR	Security Functions	Description
FIA_USB.1.1	PROC_CTRL.1	Process Control
FIA_USB.1.2	IA_UID.1	User Identification
	IA_UID.2	User Identification
	PROC_CTRL.2	Process Control
FIA_USB.1.3	PROC_CTRL.3	Process Control
	PROC_CTRL.4	Process Control
<b>FMT</b>		
FMT_MSA.1.1-A	DAC_FS.7	Discretionary Access Control for File System Objects
	DAC_FS.8	Discretionary Access Control for File System Objects
	DAC_FS.11	Discretionary Access Control for File System Objects
	DAC_IPC.6	Discretionary Access Control for IPC Objects
	DAC_IPC.7	Discretionary Access Control for IPC Objects
FMT_MSA.1.1-B	ACC_RBAC.5	Role Based Access Control
	ACC_RBAC.6	Roles Based Access Control
FMT_MSA.1.1-C	MAC_FS.6	Mandatory Access Control for File System Objects
	MAC_IPC.6	Mandatory Access Control for IPC Objects
	MAC_NET.5	Mandatory Access Control for Network Interface Objects
FMT_MSA.1.1-D	IA_ATTR.2	User Attributes
FMT_MSA.2.1	ACC_RBAC.7	Role Based Access Control
FMT_MSA.3.1-A	DAC_FS.5	Discretionary Access Control for File System Objects
	DAC_FS.6	Discretionary Access Control for File System Objects
	DAC_FS.12	Discretionary Access Control for File System Objects
	DAC_FS.13	Discretionary Access Control for File System Objects
	DAC_IPC.5	Discretionary Access Control for IPC Objects
FMT_MSA.3.2-A	DAC_FS.7	Discretionary Access Control for File System Objects
	DAC_FS.8	Discretionary Access Control for File System Objects
	DAC_FS.11	Discretionary Access Control for File System Objects
	DAC_IPC.6	Discretionary Access Control for IPC Objects
	DAC_IPC.7	Discretionary Access Control for IPC Objects
FMT_MSA.3.1-B	ACC_RBAC.2	Role Based Access Control
FMT_MSA.3.2-B	ACC_RBAC.8	Role Based Access Control
FMT_MSA.3.1-C	MAC_FS.5	Mandatory Access Control for File System Objects
	MAC_IPC.5	Mandatory Access Control for IPC Objects
FMT_MSA.3.2-C	MAC_IPC.6	Mandatory Access Control for IPC Objects
FMT_MTD.1.1-A	AUD_COLL.2	Audit Data Collection
	AUD_COLL.4	Audit Data Collection
	AUD_MTNS.2	Audit Log Files Maintenance
FMT_MTD.1.1-B	AUD_COLL.3	Audit Data Collection
	AUD_COLL.4	Audit Data Collection
	AUD_EVENT.2	Audit Events
	AUD_EVENT.4	Audit Events
	AUD_VWNG.1	Audit Logs Viewing

SFR	Security Functions	Description
FMT_MTD.1.1-C	IA_ATTR.2	User Attributes
FMT_MTD.1.1-D	IA_ATTR.2	User Attributes
	IA_PW.1	Password Selection, Generation and Encryption
FMT_MTD.1-E	SM_LOGIN.6	Login Session Management
FMT_MTD.1-F	IA_ATTR.2	User Attributes
	IA_ATTR.4	User Attributes
	IA_ATTR.5	User Attributes
	ACC_RBAC.1	Role Based Access Control
	ACC_RBAC.5	Role Based Access Control
	ACC_RBAC.6	Role Based Access Control
	ACC_RBAC.9	Role Based Access Control
FMT_MTD.3.1	ACC_RBAC.7	Role Based Access Control
FMT_REV.1.1-A	IA_ATTR.2	Identification and Authentication
	ACC_RBAC.1	Role Based Access Control
	ACC_RBAC.5	Role Based Access Control
FMT_REV.1.2-A	IA_ATTR.3	Identification and Authentication
	ACC_RBAC.3	Role Based Access Control
FMT_REV.1.1-B	DAC_FS.7	Discretionary Access Control for File System Objects
	DAC_FS.11	Discretionary Access Control for File System Objects
	DAC_IPC.6	Discretionary Access Control for IPC Objects
	DAC_IPC.7	Discretionary Access Control for IPC Objects
	MAC_FS.6	Mandatory Access Control for File System Objects
	MAC_IPC.6	Mandatory Access Control for IPC Objects
	MAC_NET.5	Mandatory Access Control for Network Interface Objects
	ACC_RBAC.8	Role Based Access Control
FMT_REV.1.2-B	DAC_FS.4	Discretionary Access Control for File System Objects
	DAC_FS.10	Discretionary Access Control for File System Objects
	DAC_IPC.4	Discretionary Access Control for IPC Objects
	MAC_FS.4	Mandatory Access Control for File System Objects
	MAC_IPC.4	Mandatory Access Control for IPC Objects
	MAC_NET.4	Mandatory Access Control for Network Interface Objects
	ACC_RBAC.10	Role Based Access Control
FMT_SAE.1.1	IA_ATTR.1	User Attributes
	IA_ATTR.2	User Attributes
FMT_SAE.1.2	IA_AUTH.7	User Attributes
	IA_AUTH.8	User Attributes
FMT_SMF.1.1	ACC_RBAC.1	Role Based Access Control
	ACC_RBAC.2	Role Based Access Control
	ACC_RBAC.5	Role Based Access Control
	ACC_RBAC.8	Role Based Access Control
	ACC_RBAC.9	Role Based Access Control
	IA_ATTR.1	User Attributes

SFR	Security Functions	Description
	IA_ATTR.2	User Attributes
	IA_ATTR.3	User Attributes
	IA_ATTR.4	User Attributes
	IA_ATTR.5	User Attributes
	IA_ATTR.6	User Attributes
	IA_ATTR.7	User Attributes
	IA_ATTR.8	User Attributes
	AUD_MTNS.1	Audit Log Files Maintenance
	AUD_MTNS.2	Audit Log Files Maintenance
	AUD_MTNS.3	Audit Log Files Maintenance
	AUD_MTNS.4	Audit Log Files Maintenance
	AUD_MTNS.5	Audit Log Files Maintenance
	AUD_MTNS.6	Audit Log Files Maintenance
	AUD_MTNS.7	Audit Log Files Maintenance
	AUD_VWNG.1	Audit Logs Viewing
	SM_LOGIN.6	Login Session Management
FMT_SMR.2.1	ACC_RBAC.1	Role Based Access Control
	ACC_RBAC.11	Role Based Access Control
FMT_SMR.2.2	ACC_RBAC.1	Role Based Access Control
FMT_SMR.2.3	ACC_RBAC.8	Role Based Access Control
<b>FPT</b>		
FPT_FLS.1.1	PROT_FUNCS.2	Protection Functions
FPT_ITC.CCOPP.1	IA_AUTH.5	Protection Functions
FPT_ITI.CCOPP.1	IA_AUTH.5	Protection Functions
FPT_RCV.1.1	PROT_FUNCS.2	Protection Functions
FPT_RCV.4.1	PROT_FUNCS.2	Protection Functions
FPT_STM.1.1	AUD_RECS.3	Audit Records
FPT_TEE.1.1	PROT_FUNCS.1	Protection Functions
FPT_TEE.1.2	PROT_FUNCS.1	Protection Functions
FPT_TST.1.1	PROT_FUNCS.1	Protection Functions
	PROT_FUNCS.2	Protection Functions
FPT_TST.1.2	PROT_FUNCS.1	Protection Functions
FPT_TST.1.3	PROT_FUNCS.1	Protection Functions
<b>FRU</b>		
FRU_PRS.1.1	RU_PRIO.1	Process Priority
	RU_PRIO.2	Process Priority
FRU_PRS.1.2	RU_PRIO.3	Process Priority
FRU_RSA.1.1	RU_QUOTA.1	Maximum Quota
	RU_QUOTA.2	Maximum Quota
	RU_QUOTA.3	Maximum Quota
	RU_QUOTA.4	Maximum Quota
FTA_LSA.1.1	SM_LOGIN.1	Login Session Management

---

Common Criteria

---

<b>SFR</b>	<b>Security Functions</b>	<b>Description</b>
FTA_MCS.1.1	SM_LOGIN.3	Login Session Management
FTA_MCS.1.2	SM_LOGIN.4	Login Session Management
FTA_SSL.4.1	SM_LOGIN.5	Login Session Management
FTA_TAB.1.1	SM_LOGIN.6	Login Session Management
FTA_TAH.1.1	SM_LOGIN.7	Login Session Management
FTA_TAH.1.2	SM_LOGIN.8	Login Session Management
FTA_TAH.1.3	SM_LOGIN.9	Login Session Management
FTA_TSE.1.1	SM_RBAC.1	RBAC Session Management
	SM_LOGIN.2	Login Session Management

## APPENDIX A: References

- [CC-V3.1]** Common Criteria for Information Technology Security Evaluation:
- Part 1 Introduction and general model  
September 2006, Version 3.1 Revision 1, CCMB-2006-09-001
  - Part 2 Security functional components  
September 2007, Version 3.1 Revision 2, CCMB-2007-09-002
  - Part 3 Security assurance components  
September 2007, Version 3.1 Revision 2, CCMB-2007-09-003
- [CCOPP-OS]** COTS Compartmentalized Operations Protection Profile – Operating Systems, version 2.0, June 19, 2008
- [CAPP]** Controlled Access Protection Profile, NSA, Version 1.d, October 8, 1999
- [ECG]** Common Criteria HP-UX 11i v3 Evaluated Configuration Guide Conformant with CCOPP-OS, HP 9000 and HP Integrity Computers, Hewlett-Packard, 2009.

**APPENDIX B: Acronyms**

<b>ACL</b>	Access Control List
<b>CC</b>	Common Criteria [for IT Security Evaluation]
<b>CCOPP</b>	COTS Compartmentalized Operation PP
<b>CDE</b>	Common Desktop Environment
<b>COTS</b>	Commercial Off The Shelf
<b>DAC</b>	Discretionary Access Control
<b>DCE</b>	Distributed Computing Environment
<b>EAL</b>	Evaluation Assurance Level
<b>IPC</b>	Inter-process Communication
<b>IT</b>	Information Technology
<b>LDAP</b>	Lightweight Directory Access Protocol
<b>MAC</b>	Mandatory Access Control
<b>NFS</b>	Network File System
<b>NIS</b>	Network Information Service
<b>NIST</b>	National Institute of Standards and Technology
<b>PP</b>	Protection Profile
<b>RBAC</b>	Role Based Access Control
<b>RPC</b>	Remote Procedure Call
<b>SAR</b>	Security Assurance Requirement
<b>SF</b>	Security Function
<b>SFP</b>	Security Function Policy
<b>SFR</b>	Security Functional Requirement
<b>SNMP</b>	Simple Network Management Protocol
<b>SOF</b>	Strength of Functions
<b>ST</b>	Security Target
<b>TOE</b>	Target of Evaluation
<b>TSC</b>	TSF Scope of Control
<b>TSF</b>	TOE Security Functions
<b>TSP</b>	TOE Security Policy
<b>UDP</b>	User Datagram Protocol