HP-UX 11i v3 Common Criteria

Security Target

Against the

Controlled Access Protection Profile (CAPP)

And the

Role Based Access Control (RBAC) Protection Profile

Prepared by:	Hewlett-Packard Development Company, L.P.
Project Manager:	Mo Mohundro
Project Document id:	ST_V1.9.doc
Version ID:	Version 1.9
Date Prepared:	December11 2007

Table of Contents

1. Introduction

1.1. ST Identification

Title:	HP-UX 11i v3 Common Criteria Security Target against the Controlled Access Protection Profile [CAPP] and the Role Based Access Control [RBAC] Protection Profile, Version 1.9 Hewlett-Packard, December 11, 2007
TOE:	HP-UX 11i v3 Operating System
CC Version:	Version 2.3
Assurance Level:	EAL4 extended and augmented with ALC_FLR.3
Registration:	<To Be Filled Upon Registration>
Keywords:	Protection Profile, role-based access, discretionary access control, separation of duties, least privilege, information protection, access control, general purpose operating system

1.2. ST Overview

1.2.1. Purpose

The purpose of this ST is to define, and specify the requirements necessary to solve the security problems that organizations encounter when trying to implement readily available operating systems (perhaps with add-on packages) to handle Controlled Access environments with specific Role-Based Access Control features, working within the same operating system.

This ST has been developed from both [CAPP] and [RBAC], which have been utilized throughout this document.

1.2.2. Scope

Type of system: This ST provides the requirements necessary to specify needs for operating systems in both stand-alone and distributed multi-user mode information systems.

Type of access: This ST assumes that the authorized access to the TOE is from **authenticated users** who have a unique identifier and are authenticated prior to being granted such access. **Role-Based Access Control (RBAC)** is a mechanism to map authenticated users to the permitted operations, by associating subjects to roles to operations on objects.

Nature of use: CAPP/RBAC conformant operating systems are suitable for the protection of information in real-world environments.

- HP-UX 11i v3 compliant Operating Systems are suitable for specifying the baseline protection requirements for information in environments where all authenticated users are either:

1) trusted to not maliciously attempt to circumvent nor by-pass access controls or
2) lack the motivation or capability for sophisticated penetration attempts.

- The Role Based Access Control (RBAC) policy is a set of rules that determines access based upon the role (e.g., PERSONNEL, MEDICAL) of the subject.

Key Assumptions: Key assumptions that apply for HP-UX 11i v3 compliant Operating Systems are –

- The Target of Evaluation (TOE, the Operating System for which requirements are being specified) is comprised of CAPP-conformant Security Functional Requirements (SFRs) as well as RBAC-conformant Access Control SFRs.
- Authenticated users recognize the need for a secure IT environment.
- Authenticated users can be reasonably trusted to correctly apply the organization's security policies in their discretionary actions.
- Competent security administration is performed.
- Business practices and policies exist to assist in the implementation and enforcement of requirements that cannot be directly or fully met by HP-UX 11i v3 compliant Operating System.

1.2.3. Summary of HP-UX 11i v3 Requirements

Assurance: HP-UX 11i v3 assurances have been selected to provide the level of confidence resulting from (1) existing best practices for Operating System development and (2) an easily-identified process for third-party evaluation. This equates, in summary, to Operating System technical countermeasures that –

- are sufficient for controlling a community of authenticated users
- can provide protection against threats of inadvertent or casual attempts to breach the system security such as an unauthorized access to the TOE by masquerading as another user
- can not be expected to provide sufficient protection against sophisticated, technical attacks such as denial-of-service attacks

Functionality: The HP-UX 11i v3 Operating System addresses these user needs –

- enforcing an access control policy between active entities (subjects) and passive objects based on subject identity and allowed actions
- providing support for controlling access based upon environmental constraints such as time-of-day
- resistance to resource depletion by providing resource allocation features
- providing mechanisms to detect insecurities
- providing mechanisms for trusted recovery in the event of most system failures or detected insecurities
- supporting these capabilities in a distributed system connected via an appropriately protected network.

HP-UX 11i v3 compliant Operating System are ***NOT*** expected to –

- totally protect against malicious abuse of authorized privileges
- adequately protect against sophisticated attacks (to include denial of service)
- provide sufficient protection against installation, operation, or administration errors

1.2.4. Strength of Environment

The assurance level is EAL4 and the minimum strength of functions is SOF-medium. The assurance requirements and the minimum strength were chosen to be consistent with that level of risk and are supported by FIA_SOS.1.

1.3. CC Conformance

This ST is CC version 2.3 Part 2 extended, and is Part 3 conformant with evaluation assurance level EAL4 augmented by ALC_FLR.3 Systematic Flaw Remediation. It is Part 2 extended as the Control Protection Profile [CAPP], which this ST is based upon, includes security functional requirements that are extensions to those found in CC version 2.3 Part 2 Security Functional Requirements.

2. TOE Description

2.1. TOE Class

HP-UX 11i v3 covers Controlled Access with RBAC operating systems in both stand-alone and networked environments. The TOEs covered by this ST permit one or more processes and attached peripheral and storage devices to be used by users to perform a variety of functions requiring controlled, shared access to processing capability and information.

The TOE will provide user services directly or serve as a platform for networked applications and will support protected communications across an appropriately protected network.

The TOE incorporates network functions but contains no network specific security requirements. Networking is covered only to the extent to which the TOE can be considered to be part of a centrally managed system that meets a common set of security requirements.

2.2. Operational Environment

The TOE supports the active entities of human users and software processes. Human users, in conjunction with system processes, are accountable for all system activities. The TOE generates processes that act on behalf of either a specific human user or a uniquely identifiable system process. A process requests and consumes resources on behalf of its unique, associated user or system process. In a networked environment, a process may invoke another process on a different system.

The TOE is intended for use in both stand-alone and networked environment and will support one or more types of communication and protocols, such as:

- Synchronous process communication; e.g., remote procedure calls (RPC)
- Asynchronous process communication; e.g., message passing using user datagram protocol (UDP)
- Network management protocols; e.g., simple network management protocol (SNMP)

A compliant TOE will support –

- Users with networked access to the TOE across a private network (that is, mechanisms operating within the TOE cooperate with mechanisms in other components to exchange information with other TOE implementations across a private network)
- Several users executing tasks on the same system concurrently
- Sharing resources, such as printer and mass storage, across a network

2.3. Evaluated Configuration

The Mission Critical Environment of HP-UX 11i v3 (also known as HP-UX 11.31) is evaluated against Controlled Access Protection Profile [CAPP] and Role Based Access Control [RBAC] Protection Profile. The evaluated configurations of the TOE are defined as follows (Refer to [ECG] for details):

- The TOE executes on any supported single 64-bit computer system from the family of HP 9000 Servers and HP Integrity Servers. On a cell-based HP 9000 and HP Integrity server, the TOE executes in any nParition configured within the server. Cell-based HP 9000 and HP Integrity 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.

- The TOE executes on a single HP 9000 Server or HP Integrity Server or on an nPartition of HP 9000 or HP Integrity Server, which may be connected to other HP 9000 Servers and HP Integrity 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 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 or by hardwire connection (e.g., implement a Cluster of HP 9000 or HP Integrity systems) or indirectly by, for example, a Wide Area Network or telephone cable to provide remote computer or network services.
- 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.CONNECT connectivity assumptions in section 3.1.3. [ECG] contains details of permitted methods of connecting to the system console.
- The TOE supports user interaction via any of the supported Shells (including the POSIX, Bourne, C, and Korn Shells).
- The TOE includes the HFS and VxFS File Systems, but excludes Online VxFS.
- 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.
- The TOE executes with CDE and X-Window disabled.

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

  a) ftp(1)
  b) rexec(1)
  c) rlogin(1)
  d) telnet(1)

- The TOE shall be installed, set up, converted to use ‘Shadow Passwords’, and operated as described in [ECG], [INSTALL], [MAN PAGES], [MSW], [REL], [README], [SDAG], and [USING].
- Boot authentication shall be enabled and auditing shall be enabled in multi-user mode, as described in [ECG].

2.4. Summary of Security Features

The main security features of HP-UX 11i v3 are:

1. auditing
2. discretionary access control (DAC), including access control lists (ACLs)
3. role based access control (RBAC)
4. user identification and authentication
5. object reuse protection

Auditing: The TOE is capable of collecting audit records for all security relevant events that occur. An authorized administrator may select the users and events for which audit record is collected from time to time.

Audit records may be viewed by an authorized administrator selectively for any period on the basis of criteria such as user name, event type and outcome (e.g. success or failure).

Facilities are provided to enable the authorized administrator to manage audit log files and to ensure that audit data is retained during abnormal conditions.

Discretionary Access Control: Except for kernel daemons that operate directly on behalf of the HP-UX 11i v3 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 discretionary access control (DAC) decisions, which control the access of subjects to objects.

The TOE implements a DAC policy, which provides 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.

The TOE implements two independent ACL mechanisms:

1. HFS ACL for the HFS File System; and
2. VxFS ACL for the VxFS File System

Role Based Access Control: The TOE implements role-based access control 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 an object. For example, a network administrator has a role that permits configuring network cards.

The system simultaneously implements DAC and role-based access control policies. Membership in a role may permit a process to temporarily assume a defined set of authorizations, privileges or other abilities to which it would not otherwise be entitled. This membership may alter, but does not substitute for, DAC enforcement for that process.

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.

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

2.5. Required Security Functionality

HP-UX 11i v3 specifies the requirements for an operating system with the security functionality listed below:

- Executing the access control policy of the imposed IT security policy
- Assigning a unique identifier to each authenticated user
- Assigning a unique identifier to each system process, including those not running on behalf of a human user (e.g., processes started at system boot-up like the Unix inetd(1M) daemon)
- Authenticating the claimed user identity before allowing any user to perform any actions other than a well-defined set of operations (e.g., use of login(1) command for the identification and the authentication purposes)
- Auditing in support of individual accountability and detection of and response to insecurity
- Enabling access authorization management; i.e., the initialization, assignment, and modification of access rights (e.g., read, write, execute) to data objects with respect to (1) active entity name or group membership and (2) environmental constraints such as time-of-day of login
- Resource allocation features providing a measure of resistance to resource depletion
- Mechanisms for detecting some insecurities
- System recovery features providing a measure of survivability in the face of system failures and insecurities

- Automated support to help in the verification of secure delivery, installation, operation, and administration

3. TOE Security Environment

3.1. Assumptions

The assumptions are fully conformant with [CAPP] and [RBAC].

3.1.1. Physical Aspects

A.ASSET [RBAC] It is also assumed that the value of the stored assets merits moderately intensive penetration or masquerading attacks. 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.LOCATE [CAPP] The processing resources of the TOE will be located within controlled access facilities which will prevent unauthorized physical access.

[RBAC] The processing resources of the TOE are located within controlled access facilities that will prevent unauthorized physical access.

A.PROTECT [CAPP] The TOE hardware and software critical to security policy enforcement will be protected from unauthorized physical modification.

[RBAC] The TOE hardware and software critical to security policy enforcement will be physically protected from unauthorized modification by potentially hostile outsiders.

3.1.2. Personnel Aspects

A.ACCESS [RBAC] 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 TOE Administrator. These roles accurately reflect the users (sic) job function, responsibilities, qualifications, and/or competencies within the enterprise.

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

[RBAC] There will be one or more competent and trustworthy individuals assigned to manage TOE security. These individuals will have sole responsibility for the following functions: (a) create and maintain roles (b) establish and maintain relationships among roles (c) Assignment and Revocation of users to roles. In addition these individuals (as ‘owners of the entire corporate data’), along with object owners will have the ability to assign and revoke object access rights to roles.

A.NO_EVIL_ADM [CAPP] The system administrative personnel are not careless, willfully negligent, or hostile, and will follow and abide by the instructions provided by the administrator documentation.

A.OWNER [RBAC] A limited set of users is given the rights to “create new data objects” and they become owners for those data objects. The organization is the owner of the rest of the information under the control of TOE.

A.COOP [CAPP] 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.

3.1.3. Connectivity Aspects

A.PEER [CAPP] 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. CAPP-conformant TOEs are applicable to networked or distributed environments only if the entire network operates under the same constraints and resides within a single management domain. There are no security requirements which address the need to trust external systems or the communications links to such systems.

A.CONNECT [CAPP] All connections to peripheral devices reside within the controlled access facilities. CAPP-conformant TOEs only address security concerns related to the manipulation of the TOE through its authorized access points. Internal communication paths to access points such as terminals are assumed to be adequately protected.

[RBAC] All connections to peripheral devices reside within the controlled access facilities.

3.2. Threats

The stated threats are fully conformant with [CAPP] and [RBAC].

3.2.1. Threats addressed by the TOE

There is no requirement by [CAPP] for statement of explicit threats countered by the TOE. The threat possibilities discussed below are addressed by [RBAC] compliant TOEs.

T.ACCESS A user may gain access to resources or perform operations for which no access rights have been granted.

The term user is used to cover those who are granted some form of legitimate access to the system, but not necessarily to all data objects or possible operations on those objects.

It is assumed that such persons may possess a wide range of technical skills and, because they have some rights of access, are minimally trusted not to attempt to subvert the system or exploit the information stored thereon. However, in view of the need for separation of function inherent in the selection of RBAC, it is assumed that there is some potential for personal gain to users from attempts to perform operations on data for which they have no authority. Some users may also be motivated by curiosity to gain access to information for which they have no authority.

Two broad categories of users are identified with respect to this threat. The first category can be assumed to have limited technical skills and only be accessing the system through application level facilities. The second category can be assumed to be granted access to programming facilities (through published APIs) with the appropriate technical skills and hence may have access to more TOE functions.

T.ENTRY An unauthorized person may gain logical access to the TOE.

The term unauthorized person is used to cover all those persons who have, or may attempt to gain, physical access to the system and its terminals but have no authority to gain logical access to the system or perform operations on its information.

3.2.2. Threats addressed by the Operating Environment

The threat possibilities discussed below must be countered in order to support the RBAC security capabilities but are not addressed by RBAC compliant TOEs. Such threats must be addressed by the operating environment.

T.OPERATE Compromise of the IT assets may occur because of improper administration and operation of the TOE.

The security offered by RBAC can be assured only to the extent that the TOE is operated correctly by system administrators and users.

Users or external threat agents may, through accidental discovery or directed search, discover inadequacies in the security administration of the TOE which permit them to gain logical access to and perform operations on its resources in breach of any permissions they may have.

Potential attackers may seek to develop methods whereby the improperly administered security functions of the TOE may be circumvented during normal operation.

T.ROLEDEV The development and assignment of user roles may be done in a manner that undermines security.

In general, roles could be developed which have an incorrect or improper combination of authorizations to perform operations on objects. In addition, users could be assigned to roles that are incommensurate with their duties, giving them either too much or too little scope of authorization.

A particular concern arises in that users could be assigned conflicting roles with respect to ‘separation of duties’. An individual user could be authorized to perform multiple operations on data objects that represent the parts of a transaction that should be separated among different individuals.

3.3. Organizational Security Policies

The organizational security policies are fully conformant with [CAPP] and [RBAC].

P.AUTHORIZED_USERS [CAPP] Only those users who have been authorized to access the information within the system may access the system.

P.NEED_TO_KNOW [CAPP] The system must limit the access to, modification of, and destruction of the information in protected resources to those authorized users which have a “need to know” for that information.

P.ACCOUNTABILITY [CAPP] The users of the system shall be held accountable for their actions within the system.

P.ACCESS [RBAC] Access rights to specific data objects are determined by the owner of the object, the role of the subject attempting access, and the implicit and explicit access rights to the object granted to the role by the object owner.

4. Security Objectives

4.1. Security Objectives for the TOE

The security objectives for the TOE are fully conformant with [CAPP] and [RBAC].

The following are the CAPP TOE IT security objectives:

O.AUTHORIZATION [CAPP] The TSF must ensure that only authorized users gain access to the TOE and its resources.

O.DISCRETIONARY_ACCESS [CAPP] The TSF must control accessed (sic) to resources based on identity of users. The TSF must allow authorized users to specify which resources may be accessed by which users.

O.AUDITING [CAPP] The TSF must record the security relevant actions of users of the TOE. The TSF must present this information to authorized administrators.

O.RESIDUAL_INFORMATION [CAPP] The TSF must ensure that any information contained in a protected resource is not released when the resource is recycled.

O.MANAGE [CAPP] The TSF must provide all the functions and facilities necessary to support the authorized administrators that are responsible for the management of TOE security.

O.ENFORCEMENT [CAPP] The TSF must be designed and implemented in a manner which ensures that the organizational policies are enforced in the target environment.

The following are the RBAC TOE IT security objectives:

O.ACCOUNT [RBAC] The TOE must ensure that all users can be held accountable for their security relevant actions.

O.ADMIN [RBAC] The TOE must provide functions to enable an authorized administrator to effectively manage the TOE and its security functions, ensuring that only authorized administrators can access such functionality.

O.AUDIT [RBAC] The TOE must provide the means of recording security relevant events in sufficient detail to help an administrator of the TOE detect attempted security violations or potential misconfiguration of the TOE security features that would leave the IT assets open to compromise.

O.DUTY [RBAC] 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.

RBAC is capable of enforcing separation of duties through roles that restrict users to a subset of operations on specific data objects.

O.ENTRY [RBAC] The TOE must prevent logical entry to it by persons or processes with no rights to access it.

O.HIERARCHICAL [RBAC] The TOE must allow hierarchical definitions of roles. Hierarchical definition of roles means the ability to define roles in terms of other roles. This saves time and allows for more convenient administration of the TOE.

O.KNOWN [RBAC] Legitimate users of the system must be identified before rights of access can be granted.

RBAC assumes that there is a finite community of known users who will be granted rights of access and that system management has authority over that user community.

O.ROLE [RBAC] 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 Environment

The following are the CAPP non-IT security objectives:

O.INSTALL [CAPP] Those responsible for the TOE must ensure that the TOE is delivered, installed, managed, and operated in a manner which maintains IT security objectives.

O.PHYSICAL [CAPP] 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 objectives.

O.CREDEN [CAPP] 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 IT security objectives.

The RBAC TOE is assumed complete and self-contained and, as such, is not dependent upon any other products to perform properly. However, certain objectives with respect to the general operating environment must be met in order to support the RBAC security capabilities.

The following are the RBAC non-IT security objectives:

O.CONNECT [RBAC] Those responsible for the TOE must ensure that no connections to outside systems or users undermine the security of IT assets.

O.INSTALL [RBAC] Those responsible for the TOE must ensure that it is delivered, installed, configured, administered, and operated in a manner which maintains IT security. This includes the definition and assignment of roles.

O.PHYSICAL [RBAC] Those responsible for the TOE must ensure that that (sic) those parts of the TOE that are critical to the security policy are protected from physical attack.

5. IT Security Requirements

The security functional requirements for the TOE are listed in Table 5-1. They comprise all of the security functional requirements taken from [CAPP] and [RBAC].

The scope of the access control mechanisms described below is consistent with that of [CAPP] and [RBAC] derived security functional requirements which are based only on user/group access permission checks.

Table 5-1 Security Functional Requirements – TOE

ST Paragraph	CAPP Paragraph	RBAC Paragraph	CC Component and Functional Elements	Name	Auditable Eevents	Objectives Addressed
					Details	
5.1.1.1	5.1.1	5.1.1	FAU_GEN.1 FAU_GEN.1.1 FAU_GEN.1.2	Audit data Generation	Start-up and shutdown of the audit functions	O.ADMIN O.AUDIT/ING
					None	
5.1.1.2	5.1.2	5.1.1	FAU_GEN.2 FAU_GEN.2.1	User Identity Association	None	O.AUDIT/ING O.ADMIN O.MANAGE
					None	
5.1.1.3	5.1.3	5.1.1	FAU_SAR.1 FAU_SAR.1.1 FAU_SAR.1.2	Audit Review	Reading	O.ADMIN O.AUDIT/ING O.MANAGE
					None	
5.1.1.4	5.1.4	5.1.1	FAU_SAR.2 FAU_SAR.2.1	Restricted Audit Review	Unsuccessful attempts to read information from the audit records	O.ADMIN O.AUDIT/ING
					None	
5.1.1.5	5.1.5	5.1.1	FAU_SAR.3 FAU_SAR.3.1	Selectable Audit Review	None	O.ADMIN O.AUDIT/ING O.MANAGE
					None	
5.1.1.6	5.1.6	5.1.1	FAU_SEL.1 FAU_SEL.1.1	Selective Audit	All modifications to the audit configuration that occur while the audit collection functions are operating	O.ADMIN O.AUDIT/ING O.MANAGE
					None	
5.1.1.7	5.1.7	5.1.1	FAU_STG.1 FAU_STG.1.1 FAU_STG.1.2	Protected Audit Trail Storage	None	O.ADMIN O.AUDIT/ING
					None	

ST Paragraph	CAPP Paragraph	RBAC Paragraph	CC Component and Functional Elements	Name	Auditable Eevents	Objectives Addressed
					Details	
5.1.1.8	5.1.8		FAU_STG.3 FAU_STG.3.1	Action in case of Possible Audit Data Loss	Actions taken due to exceeding of a threshold	O.AUDIT/ING
					None	
5.1.1.9	5.1.9		FAU_STG.4 FAU_STG.4.1	Prevention of Audit Data Loss	Actions taken due to the audit storage failure	O.AUDIT/ING O.MANAGE
					None	
5.1.2.1	5.2.1	5.1.2	FDP_ACC.1 FDP_ACC.1.1	Subset Access Control	None	O.DISCRETIONARY_ACCESS O.ENTRY
					None	
5.1.2.2	5.2.2	5.1.2	FDP_ACF.1 FDP_ACF.1.1 FDP_ACF.1.2 FDP_ACF.1.3 FDP_ACF.1.4	Security Attribute Based Access Control	All requests to perform an operation on an object covered by the SFP	O.DISCRETIONARY_ACCESS O.ENTRY
					The identity of the object.	
5.1.2.3	5.2.3		FDP_RIP.2-1 FDP_RIP2.1	Object Residual Information Protection	None	O.RESIDUAL_INFORMATION
					None	
5.1.2.4	5.2.4		FDP_RIP.2-2 FDP_RIP.2.1	Subject Residual Information Protection	None	O.RESIDUAL_INFORMATION
					None	
5.1.3.1	5.3.1	5.1.3	FIA_ATD.1 FI_ATD.1.1	User Attribute Definition	None	O.ACCOUNT O.AUTHORIZATION O.DISCRETIONARY_ACCESS O.ROLE
					None	
5.1.3.2	5.3.2		FIA_SOS.1 FIA_SOS.1.1	Verification of Secrets	Rejection or acceptance by the TSF of any tested secret	O.ACCOUNT O.AUTHORIZATION
					None	
5.1.3.3	5.3.3	5.1.3	FIA_UAU.2 FIA_UAU.2.1	User Authentication Before Any Action	All use of the authentication mechanism	O.ACCOUNT O.AUTHORIZATION O.KNOWN
					None	
5.1.3.4	5.3.4		FIA_UAU.7 FIA_UAU.7.1	Protected Authentication Feedback	None	O.ACCOUNT O.AUTHORIZATION
					None	

ST Paragraph	CAPP Paragraph	RBAC Paragraph	CC Component and Functional Elements	Name	Auditable Eevents	Objectives Addressed
					Details	
5.1.3.5	5.3.5	5.1.3	FIA_UID.2 FIA_UID.2.1	User Identification Before Any Action	All use of the authentication mechanism, including the identity provided during *successful* attempts.	O.ACCOUNT O.AUTHORIZATION O.KNOWN O.ROLE
					The origin of the attempt (e.g. terminal identification.)	
5.1.3.6	5.3.6	5.1.3	FIA_USB.1 FIA_USB.1.1-1 FIA_USB.1.1-2 FIA_USB.1.1-3	User-Subject Binding	Success and failure of binding user security attributes to a subject (e.g. success and failure to create a subject).	O.ADMIN O.AUDIT/ING O.DISCRETIONARY_ACCESS O.ROLE
					None	
5.1.4.1	5.4.1		FMT_MSA.1-1 FMT_MSA.1.1	Management Of Object Security Attributes	All modifications of the values of security object attributes	O.DISCRETIONARY_ACCESS
					None	
5.1.4.2		5.1.4	FMT_MSA.1-2 FMT_MSA.1.1	Management Of Role Security Attributes	All modifications of the values of security role attributes	O.ADMIN O.HIERARCHICAL
					None	
5.1.4.3		5.1.4	FMT_MSA.2 FMT_MSA.2.1	Secure Security Attributes	All modifications of the values of secure security attributes	O.DUTY
					None	
5.1.4.4	5.4.2	5.1.4	FMT_MSA.3 FMT_MSA.3.1 FMT_MSA.3.2	Static Attribute Initialization	Modifications of the default settings of permissive or restrictive rules. All modifications of the initial value of security attributes.	O.DISCRETIONARY_ACCESS O.ROLE
					None	
5.1.4.5	5.4.3		FMT_MTD.1-1 FMT_MTD.1.1	Management of Audit Trail	All modifications to the values of the audit Trail.	O.ACCOUNT O.ADMIN O.AUDIT/ING O.MANAGE

ST Paragraph	CAPP Paragraph	RBAC Paragraph	CC Component and Functional Elements	Name	Auditable Eevents	Objectives Addressed
					Details	
					None	
5.1.4.6	5.4.4		FMT_MTD.1-2 FMT_MTD.1.1	Management of Audited Events	All modifications to the values of the audited events.	O.ACCOUNT O.ADMIN O.AUDIT/ING O.MANAGE
					The new value of the TSF data.	
5.1.4.7	5.4.5		FMT_MTD.1-3 FMT_MTD.1.1	Management of User Attributes	All modifications to the values of the user attributes.	O.ACCOUNT O.MANAGE
					The new value of the TSF data.	
5.1.4.8	5.4.6		FMT_MTD.1-4 FMT_MTD.1.1-1 FMT_MTD.1.1-2	Management of Authentication Data	All modifications to the values of the authentication data	O.ACCOUNT O.AUTHORIZATION O.MANAGE
					None	
5.1.4.9		5.1.4	FMT_MTD.1-5 FMT_MTD.1.1	Management of TSF Data	All modifications to the values of the TSF data	O.ADMIN O.ACCOUNT O.DUTY O.HIERARCHICAL
					None	
5.1.4.10		5.1.4	FMT_MTD.3 FMT_MTD.3.1	Secure TSF Data	All modifications to the values of the secure TSF data	O.ACCOUNT
					None	
5.1.4.11	5.4.7	5.1.4	FMT_REV.1-1 FMT_REV.1.1 FMT_REV.1.2	Revocation of User Attributes	All attempts to revoke user attributes	O.MANAGE
					None	
5.1.4.12	5.4.8		FMT_REV.1-2 FMT_REV.1.1 FMT_REV.1.2	Revocation of Object Attributes	All attempts to revoke object attributes	O.DISCRETIONARY_ACCESS
					None	
5.1.4.13	New	New	FMT_SMF.1 FMT_SMF.1.1	Specification of Management Functions	All attempts to utilize management functions	O.DUTY O.HIERARCHICAL O.MANAGE O.ACCOUNT
					None	
5.1.4.14	5.4.9	5.1.4	FMT_SMR.2 FMT_SMR.1.1 FMT_SMR.1.2 FMT_SMR.2.1 FMT_SMR.2.2 FMT_SMR.2.3	Security Roles and Restriction on Security Roles	Every use of the rights of a role	O.DUTY O.HIERARCHICAL O.MANAGE O.ACCOUNT
					The role and the origin of the request.	

ST Paragraph	CAPP Paragraph	RBAC Paragraph	CC Component and Functional Elements	Name	Auditable Eevents	Objectives Addressed
					Details	
5.1.5.1	5.5.1	5.1.5	FPT_AMT.1 FPT_AMT.1.1	Abstract Machine Testing	Execution of the tests of the underlying machine and the results of the tests.	O.ENTRY O.ENFORCEMENT
					None	
5.1.5.2		5.1.5	FPT_FLS.1 FPT_FLS.1.1	Failure with Preservation of Secure State	The ability of the system to return to a benign state after failure.	O.ENTRY
					None	
5.1.5.3		5.1.5	FPT_RCV.1 FPT_RCV.1.1	Manual Recovery	The ability to recover manually from failure	O.ADMIN
					None	
5.1.5.4		5.1.5	FPT_RCV.4 FPT_RCV.4.1	Function recovery	The ability of the security functions to either complete or fail to a benign state	O.ROLE
					None	
5.1.5.5	5.5.2	5.1.5	FPT_RVM.1 FPT_RVM.1.1	Non-Bypassability of the TSP	None	O.ENFORCEMENT O.ENTRY
					None	
5.1.5.6	5.5.3	5.1.5	FPT_SEP.1 FPT_SEP.1.1 FPT_SEP.1.2	TSF Domain Separation	None	O.ENFORCEMENT O.ENTRY
					None	
5.1.5.7	5.5.4	5.1.5	FPT_STM.1 FPT_STM.1.1	Reliable Time Stamps	Changes to the time	O.ADMIN O.AUDIT/ING
					None	
5.1.5.8		5.1.5	FPT_TST.1 FPT_TST.1.1 FPT_TST.1.2 FPT_TST.1.3	TSF Testing	Testing to ensure correct operation of TSF	O.ACCOUNT
					None	
5.1.6.1		5.1.6	FTA_LSA.1 FTA_LSA.1.1	Limitation on Scope of Selectable Attributes	Attributes are only selectable by Role	O.ENTRY
					None	
5.1.6.2		5.1.6	FTA_TSE.1 FTA_TSE.1.1	TOE Session Establishment	Deniability of session establishment by Role	O.ENTRY

ST Paragraph	CAPP Paragraph	RBAC Paragraph	CC Component and Functional Elements	Name	Auditable Eevents	Objectives Addressed
					Details	
					None	

5.1. TOE Security Functional Requirements

This section provides the definitions of the security functional requirements for the TOE drawn from the Part 2 of CC v2.3, [CAPP] and [RBAC]. Operations performed on functional components are highlighted as described in Section 7.2 PP Tailoring.

5.1.1. Security Audit (FAU)

5.1.1.1. Audit Data Generation (FAU_GEN.1)

FAU_GEN.1.1

[CAPP 5.1.1.1] The TSF shall be able to generate an audit record of the auditable events *listed in column "Event" of* ***Table 5-1(Security Functional Requirements – TOE)****. This includes all auditable events for the basic level of audit, except FIA_UID.1's user identity during failures.*

[RBAC 5.1.1] The TSF shall be able to generate an audit record of the following auditable events:

a) Start-up and Shutdown of the audit functions;
b) All auditable events for the *basic* level of audit; and
c) i) *Assignment of Users, Roles and Privileges to Roles*
ii) *Deletion of Users, Roles and Privileges from Roles*
iii) *Creation and Deletion of Roles*

FAU_GEN.1.2

[CAPP 5.1.1.2] 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, and the outcome (success or failure) of the event;

b) *The additional information specified in the "Details" column of* ***Table 5-1(Security Functional Requirements – TOE)****.*

[RBAC 5.1.1] 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, and the outcome (success or failure) of the event; and

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST the following information ***(see "Auditable Events" column of Table 5-1)***:

i) *For each invocation of a security function, the RBAC Administrator role that made invocation of that security function possible*.

ii) *For each access control action on the user data, the role that made possible the invocation of that action.*

5.1.1.2. User Identity Association (FAU_GEN.2)

FAU_GEN.2.1

[CAPP 5.1.2.1] [RBAC 5.1.1] 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

[CAPP 5.1.3.1] The TSF shall provide *authorized administrators* with the capability to read *all audit information* from the audit records.

[RBAC.5.1.1] The TSF shall provide *the set of authorized RBAC administrators* with the capability to read *the following audit information* from the audit records:

a) *Date and Time of Audit Event*
b) *The UserID responsible for the Event and optionally the role membership which enabled the user to perform the event successfully*
c) *The access control operation and the object on which it was performed*
d) *The outcome of the event (success or failure)*
e) *The User Session Identifier or Terminal Type*

FAU_SAR.1.2

[CAPP 5.1.3.2] [RBAC 5.1.1] 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

[CAPP 5.1.4.1] [RBAC 5.1.1] 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

[CAPP 5.1.5.1] The TSF shall provide the ability to perform ***searches*** of audit data based on *the following attributes*:

a) *User Identity*;
b) ***Terminal port***;
c) ***Set of event types***;
d) ***Set of system calls***;
e) ***Successful events***;
f) ***Failed events***;
g) ***The date and time, or period, in which the event occurred.***

[RBAC 5.1.1] The TSF shall provide the ability to perform *searches, sorting and ordering* of audit data based on *the following criteria*:

a) *Date and Time of Audit Event*
b) *UserID*
c) *Object Name & type of access*
d) *Role that enabled the access*
e) *Any combination of the above items (a), (b), (c) or (d).*

5.1.1.6. Selective Audit (FAU_SEL.1)

FAU_SEL.1.1

[CAPP 5.1.6.1] The TSF shall be able to include or exclude auditable events from the set of audited events based on the following attributes:

a) *User Identity*

[RBAC 5.1.1] The TSF shall be able to include or exclude auditable events from the set of audit events based on the following attributes:

a) *Object identity, user identity, subject identity, host identity, and event type*
b) *Users belonging to a specified Role and Access types (e.g., delete, insert) on a particular object.*

5.1.1.7. Protected Audit Trail Storage (FAU_STG.1)

FAU_STG.1.1

[CAPP 5.1.7.1] [RBAC 5.1.1] The TSF shall protect the stored audit records from unauthorized deletion.

FAU_STG.1.2

[CAPP 5.1.7.2] [RBAC 5.1.1] The TSF shall be able to *prevent* modification to the audit records

5.1.1.8. Actions in Case of Possible Audit Data Loss (FAU_STG.3)

FAU_STG.3.1

[CAPP 5.1.8.1] The TSF shall *generate an alarm to the authorized administrator* if the audit trail exceeds ***an authorized administrator's configurable percentage of the storage capacity***.

5.1.1.9. Prevention of Audit Data Loss (FAU_STG.4)

FAU_STG.4.1

[CAPP 5.1.9.1] The TSF shall *be able to prevent auditable events, except those taken by authorized administrator*, if the audit trail is full.

5.1.2. User Data Protection (FDP)

5.1.2.1. Subset Access Control (FDP_ACC.1)

FDP_ACC.1.1

[CAPP 5.2.1.1] The TSF shall enforce the *Discretionary Access Control Policy* on ***all subjects*** *acting on behalf of users*, ***File System, System V IPC and POSIX IPC objects*** *and all operations among subjects and objects covered by the DAC policy*.

[RBAC 5.1.2] The TSF shall enforce the *Role-based Access Control (RBAC) SFP* on:

a) *Subjects (specified in the RBAC ST) covered by RBAC SFP*
b) *Objects (specified in the RBAC ST) covered by RBAC SFP*
c) *All Operations on Objects (specified in RBAC ST) covered by RBAC SFP*

5.1.2.2. Security Attribute Based Access Control (FDP_ACF.1)

FDP_ACF.1.1

[CAPP 5.2.2.1] The TSF shall enforce the *Discretionary Access Control 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 HFS File System Objects, the Access Mode Permissions and the HFS ACL;***
    ii) ***For VxFS File System Objects, the Access Mode Permissions and the VxFS ACL;***
    iii) ***For System V IPC and POSIX IPC Objects, the Access Mode Permissions.***

[RBAC 5.1.2 (1)] The TSF shall enforce the *RBAC SFP* to objects based on the following *user attributes*:

a) *User Identity*
b) *Authorized Roles for the User*

[RABC 5.1.2 (2)] The TSF shall enforce the *RBAC SFP* to objects based on the following *subject attributes*:

a) *Subject Identity*
b) *Role(s) which can invoke the subject*

[RABC 5.1.2 (3)] The TSF shall enforce the *RBAC SFP* to objects based on the following *object attributes*:

a) *Object Identity*
b) *Operations permitted on the objects for various Roles*

FDP_ACF.1.2

[CAPP 5.2.2.2] 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 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 effective group 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 in the user: : entry if there is a match with the object's owner class of user*;**

iii) ***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*.**

iv) ***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*;**

v) ***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*;**

vi) ***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)*:**

ii) ***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*;**

iii) ***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*;**

iv) ***Access is granted or denied according to the permissions of the object's other class of user*.**

[RBAC 5.1.2] The TSF shall enforce the following rules to determine if any operation among controlled subjects and controlled objects is allowed:

a) *The subject invoking the operation on an object is assigned to a role whose privilege set includes the operation on the object*.

FDP_ACF.1.3

[CAPP 5.2.2.3] The TSF shall explicitly authorize access of subjects to objects based on the following additional rules:

a) ***An authorized administrator shall be granted access to all objects, overriding the rules specified in FDP_ACF.1.2*.**

[RBAC 5.1.2] The TSF shall explicitly authorize access of subject to objects based on the following additional rules:

a) *Allow an access operation by a subject on an object only if the user associated with the subject belongs to a role that permits the access operation on the object*.

FDP_ACF.1.4

[CAPP 5.2.2.4] The TSF shall explicitly deny access of subject to objects based on ***no other rules than those specified in FDP_ACF.1.2.***

[RBAC 5.1.2] The TSF shall explicitly deny access of subjects to objects based on the *user associated with the subject not belonging to any role that permits the requested access operation on the object*.

5.1.2.3. Object Residual Information Protection (FDP_RIP.2-1)

FDP_RIP.2.1

[CAPP 5.2.3.1] The TSF shall ensure that any previous information content of a resource is made unavailable upon the *allocation of the resource* to all objects.

5.1.2.4. Subject Residual Information Protection (FDP_RIP.2-2)

FDP_RIP.2.1

[CAPP 5.2.4.1] 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. User Attribute Definition (FIA_ATD.1)

FIA_ATD.1.1

[CAPP 5.3.1.1] The TSF shall maintain the following list of security attributes belonging to individual users:

a) *User Identifier*;
b) *Group Membership*;
c) *Authentication Data*;
d) *Security-relevant Roles*; and
e) ***Audit tag (session specific)*;**
f) ***Home directory*;**
g) ***Login program*;**
h) ***Audit flag; and***
i) ***Boot flag*;**

[RBAC 5.1.3] The TSF shall maintain the following list of security attributes belonging to individual users:

a) *List of Authorized Roles*;
b) *Any other user attributes related to Roles, as defined in the RBAC ST*.

5.1.3.2. Strength of Authentication Data (FIA_SOS.1)

FIA_SOS.1.1

[CAPP 5.3.2.1] 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*.

5.1.3.3. User Authentication before Any Action (FIA_UAU.2)

FIA_UAU.2.1

[RBAC 5.1.3] The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

5.1.3.4. Protected Authentication Feedback (FIA_UAU.7)

FIA_UAU.7.1

[CAPP 5.3.4.1] The TSF shall provide only *obscured feedback* to the user while the authentication is in progress.

5.1.3.5. User Identification Before Any Action (FIA_UID.2)

FIA_UID.2.1

[RBAC 5.1.3] The TSF shall require each user to identify itself before allowing any other TSF-mediated actions on behalf of that user.

5.1.3.6. User-Subject Binding (FIA_USB.1)

FIA_USB.1.1-1

[CAPP 5.3.6.1] 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*; ***and***

d) ***The current working directory*;**

[RBAC 5.1.3] The TSF shall associate the appropriate user security attributes with subjects acting on behalf of the user.

FIA_USB.1.1-2

[CAPP 5.3.6.2] *The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of a user*:

a) ***The user identifier which is associated with auditable events is initialized to the audit tag appropriate to that user's identity and session parameters*.**

b) ***The user identity or identities which are used to enforce the Discretionary Access Control Policy are set to the User Identifier*;**

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 group access list used to enforce the Discretionary Access Control Policy are set to the user's supplementary Group Memberships*;**

e) ***The current working directory is set to the user's home directory*.**

FIA_USB.1.1-3

[CAPP 5.3.6.3] *The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of a user*:

a) ***An authorized administrator shall be able to change the user identities and group memberships of a subject acting on his behalf to that of another valid user (the su(1) command)*;**

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; and***

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.

5.1.4. Security Management (FMT)

5.1.4.1. Management of Object Security Attributes (FMT_MSA.1-1)

FMT_MSA.1.1

[CAPP 5.4.1.1] The TSF shall enforce the *Discretionary Access Control Policy* to restrict the ability to *modify* the *access control 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 CHOWN 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****.*

[RBAC 5.1.4] The TSF shall enforce the *RBAC SFP* to restrict the ability to *modify* the object security attributes to

i) ***Object Owners and***

ii) ***Set of RBAC administrative roles****.*

5.1.4.2. Management of Role Security Attributes (FMT_MSA.1-2)

FMT_MSA.1.1

[RBAC 5.1.4 (1)] The TSF shall enforce the *RBAC SFP* to restrict the ability to *modify, delete, and create instances* of the *following user* security attributes to *a set of RBAC Administrative Roles*:

a) *User Role Authorizations*

[RBAC 5.1.4 (2)] The TSF shall enforce the *RBAC SFP* to restrict the ability to *create and modify the composition* of the *following user* security attribute to *a set of RBAC Administrative Roles*:

a) *Default Active Role Set*

[RBAC 5.1.4 (3)] The TSF shall enforce the *RBAC SFP* to restrict the ability to *modify the composition* of the *following session* security attribute to *session owner*:

a) *Active Role Set for a user*

5.1.4.3. Secure Security Attributes (FMT_MSA.2)

FMT_MSA.2.1

[RBAC 5.1.4] The TSF shall ensure that only secure values are accepted for security attributes.

5.1.4.4. Static Attribute Initialization (FMT_MSA.3)

FMT_MSA.3.1

[CAPP 5.4.2.1] The TSF shall enforce the *Discretionary Access Control Policy* to provide *restrictive* default values for security attributes that are used to enforce the Discretionary Access Control Policy.

[RBAC 5.1.4] The TSF shall enforce the *RBAC SFP* to provide default values for object security attributes that are used to enforce the SFP.

FMT_MSA.3.2

[CAPP 5.4.2.2] The TSF shall allow the ***authorized administrator 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.

[RBAC 5.1.4] The TSF shall allow the *following* roles to specify alternative initial values to override the default values when an object or information is created:

a) *Set of RBAC Administrative Roles*

5.1.4.5. Management of Audit Trail (FMT_MTD.1-1)

FMT_MTD.1.1 (1)

[CAPP 5.4.3.1] The TSF shall restrict the ability to *create, delete, and clear* the *audit trail* to *authorized administrators*.

5.1.4.6. Management of Audited Events (FMT_MTD.1-2)

FMT_MTD.1.1 (2)

[CAPP 5.4.4.1] The TSF shall restrict the ability to *modify or observe* the *set of audited events* to *authorized administrators*.

5.1.4.7. Management of User Attributes (FMT_MTD.1-3)

FMT_MTD.1.1 (3)

[CAPP 5.4.5.1] The TSF shall restrict the ability to *initialize and modify* the *user security attributes, other than authentication data,* to *authorized administrators*.

5.1.4.8. Management of Authentication Data (FMT_MTD.1-4)

FMT_MTD.1.1 (4)

[CAPP 5.4.6.1] The TSF shall restrict the ability to *initialize* the *authentication data* to *authorized administrators*.

FMT_MTD.1.1 (4)

[CAPP 5.4.6.2] 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*

5.1.4.9. Management of TSF Data (FMT_MTD.1-5)

FMT_MTD.1.1 (5)

[RBAC 5.1.4] The TSF shall restrict the ability to *modify or create* the *following list of TSF data* to *a set of RBAC Administrative Roles*:

a) *All User Passwords*
b) *Role Definitions & Role Attributes*
c) *Role Hierarchies (by assigning one or more roles to other roles)*
d) *Constraints among Role Relationships*
e) *List of Auditable Events*

5.1.4.10. Secure TSF Data (FMT_MTD.3)

FMT_MTD.3.1

[RBAC 5.1.4] The TSF shall ensure that only secure values are accepted for TSF data.

5.1.4.11. Revocation of User Attributes (FMT_REV.1-1)

FMT_REV.1.1

[CAPP 5.4.7.1] The TSF shall restrict the ability to revoke security attributes associated with the *users* within the TSC to *authorized administrators*.

FMT_REV.1.2

[CAPP 5.4.7.2] 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 trusted user to log off.

Note: The stated FMT_REV.1 SFRs also comply with [RBAC] as shown in Section 8.4.

5.1.4.12. Revocation of Object Attributes (FMT_REV.1-2)

FMT_REV.1.1

[CAPP 5.4.8] The TSF shall restrict the ability to revoke security attributes associated with *objects* within the TSC to ***<u>the following users</u>*** *authorized to modify the security attributes by the Discretionary Access Control Policy*:

a) ***<u>Object Owners and</u>***
b) ***<u>Authorized Administrators</u>***

FMT_REV.1.2

[CAPP 5.4.8.1] The TSF shall enforce the rules:

a) *The access rights associated with an object shall be enforced when an access check is made*.

Note: The stated FMT_REV.1 SFRs also comply with [RBAC] as shown in Section 8.4. SFR FMT_REV.1.1 is refined to also comply with [RBAC].

5.1.4.13. Specification of Management Functions (FMT_SMF.1)

FMT_SMF.1.1

The TSF shall be capable of performing the following security management functions:

a) ***Start and halt the auditing system***
b) ***Select users and events to be audited***
c) ***Add, modify, and delete user profiles***
d) ***Add, modify, assign, and delete user roles***
e) ***Add, modify, assign, and delete authorizations to users and processes***

5.1.4.14. Restriction on Security Roles (FMT_SMR.2)

FMT_SMR.2.1

[CAPP 5.4.9.1 (FMT_SMR1.1)] [RBAC 5.1.4 (FMT_SMR_2.1)] The TSF shall maintain the roles:

a) *authorized administrators (CAPP 5.4.9.1)*;

b) *users authorized by the Discretionary Access Control Policy to modify object security attributes (CAPP 5.4.9.1)*;

c) *users authorized to modify their own authentication data (CAPP 5.4.9.1)*;

d) *set of RBAC administrative roles (RBAC 5.1.4)*;

e) *roles for Object Owners (RBAC 5.1.4);* and

f) ***users authorized by the Role Based Access Control Policy to modify object security attributes***.

FMT_SMR.2.2

[RBAC 5.1.4] The TSF shall be able to associate users with roles.

FMT_SMR.2.3

[RBAC 5.1.4] 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. Abstract Machine Testing (FPT_AMT.1)

FPT_AMT.1.1

[CAPP 5.5.1.1] The TSF shall run a suite of tests ***at the request of an authorized administrator*** to demonstrate the correct operation of the security assumptions provided by the abstract machine that underlies the TSF.

[RBAC 5.1.5] The TSF shall run a suite of tests *periodically during normal operation and at the request of an authorized user* to demonstrate the correct operation of the security assumptions provided by the abstract machine that underlies the TSF.

5.1.5.2. Failure with Preservation of Secure State (FPT_FLS.1)

FPT_FLS.1.1

[RBAC 5.1.5] The TSF shall preserve a secure state when the *following 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*

5.1.5.3. Manual Recovery (FPT_RCV.1)

FPT_RCV.1.1

[RBAC 5.1.5] After *a failure or service discontinuity,* the TSF shall enter a maintenance mode where the ability to return the TOE to a secure state is provided.

5.1.5.4. Function Recovery (FPT_RCV.4)

FPT_RCV.4.1

[RBAC 5.1.5] The TSF shall ensure that *the following SFs and failure scenarios* have the property that the SF either completes successfully, or for the indicated failure scenarios, recovers to a consistent and secure state:

a) *The SF that checks whether a specified privilege is assigned to any role but the database containing the privilege data is not on-line or the particular data table is inaccessible.*

b) *The SF that checks whether a specified role has been assigned to a particular user but the database containing the role membership information is not on-line or the particular data table is inaccessible.*

5.1.5.5. Non-bypassability of the TSP (FPT_RVM.1)

FPT_RVM.1.1

[CAPP 5.5.2.1] [RBAC 5.1.5] The TSF shall ensure that the TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.

5.1.5.6. TSF Domain Separation (FPT_SEP.1)

FPT_SEP.1.1

[CAPP 5.5.3.1] [RBAC 5.1.5] The TSF shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects.

FPT_SEP.1.2

[CAPP 5.5.3.2] [RBAC 5.1.5] The TSF shall enforce separation between the security domains of subjects in the TSC.

5.1.5.7. Reliable Time Stamps (FPT_STM.1)

FPT_STM.1.1

[CAPP 5.5.4.1] [RBAC 5.1.5] The TSF shall be able to provide reliable time stamps for its own use.

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.

5.1.5.8. TSF Testing (FPT_TST.1)

FPT_TST.1.1

[RBAC 5.1.5] The TSF shall run a suite of self tests ***at the request of the authorized user*** to demonstrate the correct operation of the TSF.

FPT_TST1.2

[RBAC 5.1.5] The TSF shall provide authorized users with the capability to verify the integrity of *TSF data*.

FPT_TST.1.3

[RBAC 5.1.5] The TSF shall provide authorized users with the capability to verify the integrity of stored executable code.

5.1.6. TOE Access (FTA)

5.1.6.1. Limitation on Scope of Selectables Attributes (FTA_LSA.1)

FTA_LSA.1.1

[RBAC 5.1.6] The TSF shall restrict the scope of the session security attributes (*Active Role Set for the User*) based on *the set of Authorized Roles for the User*.

5.1.6.2. TOE Session Establishment (FTA_TSE.1)

FTA_TSE.1.1

[RBAC 5.1.6] The TSF shall be able to deny session establishment based on *the default active role set for the user being empty*.

5.2. Strength of Function

The claimed strength of function is SOF-Medium, and is supported by FIA_SOS.1. The only mechanism that has the SOF requirement is the password.

5.3. TOE Security Assurance Requirements

The target evaluation assurance level for HP-UX 11i v3 is EAL4, augmented by ALC_FLR.3 Systematic Flaw Remediation.

5.4. Security Requirements for the IT Environment

There are no [CAPP] security requirements for the IT environment. [RBAC] security environmental requirements are overcome with the [CAPP] inclusion.

6. TOE Summary Specification

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) an audit tag (dynamically assigned at session creation)
h) the current working directory

A user gains initial access to the TOE via login at a terminal, which involves authentication of the user. A successful login results in the creation of a 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 audit tag is dynamically generated and assigned.
e) 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 and group IDs 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 possess appropriate privileges. The appropriate privileges must be associated with the process that is performing the action on behalf of the user.

The TOE provides the following types of standard privilege:

a) authorized administrator status, that is, a process executing with an effective user ID of zero, equivalent to the root user

b) a system capability associated with privilege groups, that is, a process executing with an effective group ID or group access list which includes a group that has been given one or more system capabilities

A process with authorized administrator status is not constrained by the TOE's security policies.

A process may possess the CHOWN security relevant system capability, 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.

6.1.1.2. Objects and Access Permissions

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.

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.

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

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

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.

6.1.1.3. Security Policy Rationales

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 users control over “access to objects” under their management
b) to protect user activities from undesired interference.

6.1.1.4. Role Based Access Control (RBAC) Mechanism

In addition to the standard access control mechanism described in sections 6.1.1.1 and 6.1.1.2, the TOE also 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. 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

6.1.1.5. 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.

6.1.2. Probablistic and Permutational Mechanisms

The only probabilistic mechanism that is used by HP-UX 11i v3 is the password.

The strength of function (SOF) of the passwords is medium, which is defined in FIA_SOS.1. The requirements imposed on the password by the SOF are met as specified in security function PW_SEL&GEN.2 and as stated by the following implementation mechanisms (excluding superuser or user id 0):

a. Each password shall have at least six characters long and no more than eight;
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; and
d. New passwords shall differ from the old password by at least three characters.
e. The password checking algorithm which enforces the constraints on user-generated passwords should satisfy the Strength of Function claim of SOF-medium.
f. A modified one-way DES algorithm is implemented to satisfy the password encryption function specified in security function PW_ENCR.1.

The assessment of the Strength of Function of encryption algorithms is outside the scope of the evaluation.

6.1.3. SFR to SF Mapping

Table 8-5 in Section 8.3 shows the mapping of security functional requirements to security functions.

6.2. TOE Security Functions

6.2.1. Audit (AUD)

6.2.1.1. Audit Data Collection (AUD_DATA_COLL)

AUD_DATA_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_DATA_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_DATA_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_DATA_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_DATA_COLL.5 The TOE shall log start-up and shut-down of the auditing functions.

AUD_DATA_COLL.6 If the RBAC mechanism is activated, the TOE shall audit records for the creation, assignment, modification, and deletion of roles, role authorizations, and command authorizations. This is accomplished as per self-auditing functions roleadm(1M), authadm(1M), and cmdprivadm(1M).

6.2.1.2. Audit Events (AUD_EVENTS)

AUD_EVENTS.1 The TOE shall group system calls having a similar behavior into categories called ‘event types’.

AUD_EVENTS.2 The TOE shall provide the event types listed in the table below.

AUD_EVENTS.3 The TOE shall allow only an authorized administrator to set or observe the auditing status of event types, on a per event type basis, to one of the following:

AUD_EVENTS.3.1 audit for success only
AUD_EVENTS.3.2 audit for failure only
AUD_EVENTS.3.3 audit for both success and failure
AUD_EVENTS.3.4 do not audit.

AUD_EVENTS.4 The TOE shall allow only an authorized administrator to set or observe the auditing status of system calls, on a per system call basis, to one of the following;

AUD_EVENTS.4.1 audit for success only
AUD_EVENTS.4.2 audit for failure only
AUD_EVENTS.4.3 audit for both success and failure
AUD_EVENTS.4.4 do not audit

AUD_EVENTS.5 The TOE’s initial default selection of audit shall audit the success and failure of the following event types:

AUD_EVENTS.5.1 admin
AUD_EVENTS.5.2 logon
AUD_EVENTS.5.3 moddac

Table 6-1 Audit Event Types and System Calls

Audit 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), *getksym*(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), *setaudid*(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	*close*(2), *ksem_close*(2), *mq_close*(2), *munmap*(2)

Audit Event Types and System Calls		
Event Type	**Description of Action**	**Associated System Calls**
	objects	
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)
ipcopen	Log all ipc open events	*accept*(2), *connect*(2), *fatach*(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), *vfsmount*(2)
uevent1 uevent2 uevent3	Log user defined events	See AUD_LOG_DATA_STRMG, section 6.2.1.3

6.2.1.3. Audit Log Data Streamlining (AUD_LOG_DATA_STRMG)

AUD_LOG_DATA_STRMG.1 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_LOG_DATA_STRMG.2 The process listed in the table below shall be self-auditing.

AUD_LOG_DATA_STRMG.3 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_EVENTS.3:

AUD_LOG_DATA_STRMG.3.1	uevent1
AUD_LOG_DATA_STRMG.3.2	uevent2
AUD_LOG_DATA_STRMG.3.3	uevent3

Table 6-2 Self-Auditing Processes

Self-auditing Processes	
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.
privrun(1M)	Executive component of RBAC. Execute a legacy process after performing appropriate authorization check
roleadm(1M)	Administers role-related information in RBAC databases
useradd(1M)	Add new user login account
userdel(1M)	Delete user login account
usermod(1M)	Modify user login account

6.2.1.4. Audit Records (AUD_RECS)

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

AUD_RECS.1.1 process ID
AUD_RECS.1.2 parent process ID
AUD_RECS.1.3 audit tag
AUD_RECS.1.4 real user ID
AUD_RECS.1.5 real group ID
AUD_RECS.1.6 effective user ID
AUD_RECS.1.7 effective group ID
AUD_RECS.1.8 device name

AUD_RECS.2 For each event audited, the TOE shall record in the selected audit log file the following information:

AUD_RECS.2.1 the system date and time that the audited event compeletes
AUD_RECS.2.2 the event type
AUD_RECS.2.3 the process ID of the process that causes the event
AUD_RECS.2.4 the success or failure of the event.
AUD_RECS.2.5 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 If the RBAC mechanism is activated, for each RBAC audit event type, the TOE shall record the following information:

AUD_RECS.6.1 Role assigned to user
AUD_RECS.6.2 The role authorization
AUD_RECS.6.3 The process authorization
AUD_RECS.6.4 The operation performed (process)
AUD_RECS.6.5 The object on which the operation was performed

6.2.1.5. Audit Logs Viewing (AUD_LOGS_VWNG)

AUD_LOG_VWNG.1 The TOE shall provide the capability for only the authorized administrator to extract audit log data (see sections 6.2.1.1 and 6.2.1.4) from a specified audit log file in accordance with one or more or the following selection criteria:

AUD_LOG_VWNG.1.1 a given user name

AUD_LOG_VWNG.1.2 a given terminal name

AUD_LOG_VWNG.1.3 a given set of event types

AUD_LOG_VWNG.1.4 a given set of system calls

AUD_LOG_VWNG.1.5 successful events

AUD_LOG_VWNG.1.6 failed events

AUD_LOG_VWNG.1.7 the event date and time at which to start the extraction of audit log data

AUD_LOG_VWNG.1.8 the event date and time at which to end the extraction of audit log data

AUD_LOG_VWNG.2 If the RBAC mechanism is activated, the TOE shall provide the capability of defining a role for an authorized (RBAC) administrator to extract audit log data (see sections 6.2.1.1 and 6.2.1.4) in accordance with the selection criteria defined in AUD_LOG_VWNG.1 as well as the following selection criteria:

AUD_LOG_VWNG.2.1 the role that enables the access

AUD_LOG_VWNG.2.2 object name associated with the event

AUD_LOG_VWNG.2.3 operation performed on the object

AUD_LOG_VWNG.2.4 any combination of above-mentioned items

6.2.1.6. Audit Log Files Maintenance (AUD_LOG_FILES_MTNS)

AUD_LOG_FILES_MTNS.1 The TOE shall collect audit records in:

AUD_LOG_FILES_MTNS.1.1 a *primary* log file, which is used initially by the TOE

AUD_LOG_FILES_MTNS.1.2 an optional *auxiliary* log file selected by an authorized administrator

AUD_LOG_FILES_MTNS.2 The TOE shall allow authorized administrator to specify the following parameters:

AUD_LOG_FILES_MTNS.2.1 an Audit File Switch (AFS) size

AUD_LOG_FILES_MTNS.2.2 the File Space Switch (FSS) size

AUD_LOG_FILES_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_LOG_FILES_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_LOG_FILES_MTNS.5 If no auxiliary log file exists, the TOE shall periodically issue a warning on the console.

AUD_LOG_FILES_MTNS.6 When the space available on the file system(s) containing the primary log file and the auxiliary log file is exhausted, all auditable actions of unprivileged users shall be suspended.

AUD_LOG_FILES_MTNS.7 When the file system(s) is (are) completely full, no audit records shall be collected, although an authorized administrator shall be allowed to continue to carry out operations.

AUD_LOG_FILES_MTNS.8 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.

6.2.2. User Data Protection (FDP)

6.2.2.1. Discretionary Access Control (ACC_DAC)

ACC_DAC.1 The TOE shall define and control discretionary access between subjects and objects. (See seciton 6.1.1. for a definition of subjects and objects).

ACC_DAC.2 The TOE's definition and control of discretionary access between subjects and objects shall be implemented by the following discretionary access control (DAC) mechanisms:

ACC_DAC.2.1 access mode (owner/group/other) permissions
ACC_DAC.2.2 access control lists (ACLs)

ACC_DAC.3 ACLs shall only be applied to File System Objects, as follows:

ACC_DAC.3.1 for HFS File Systems using an HFS ACL
ACC_DAC.3.2 for VxFS File Systems using a VxFS ACL

6.2.2.2. Role Based Access Control (ACC_RBAC)

ACC_RBAC.1 The TOE shall use "roleadm(1m)", "authadm(1m)", and "cmdprivadm(1M)" to assign, modify, and revoke roles to users, assign, modify, and revoke authorizations to roles, and assign, modify, and revoke authorizations to other commands/processes.

ACC_RBAC.2 The "authadm(1M)" command assumes default values for the object security attributes when not specified.

ACC_RBAC.3 The assignment/revocation of roles and authorizations will take effect immediately.

ACC_RBAC.4 The RBAC SFP is enforced through Access Control Policy (ACPS) subsystem by verifying user, role, and process authorizations before allowing or denying the operation to take place on the object (See 6.1.1.4 for more details).

ACC_RBAC.5 The TOE is capable of defining an (RBAC) administrator role to create, modify, and delete the following user security attributes.

ACC_RBAC.5.1 User Role Authorization
ACC_RBAC.5.2 Default Active Role Set

ACC_RBAC.6 The TOE is capable of restricting the ability to modify the following session security attribute to (RBAC) administrator role and session owner:

ACC_RBAC.6.1 Active Role set for a user

ACC_RBAC.7 The TOE shall use a two step process to a) assign a role to a user (roleadm(1m) and b) assign authorization to a role (authadm(1M)). The two steps process ensures that acceptable values are assigned to security attributes.

ACC_RBAC.8 Only object owners and an authorized (RBAC) administrator are able to modify and/or revoke object security attributes.

ACC_RBAC.9 Role hierarchies are supported in the TOE RBAC mechanism.

ACC_RBAC.10 The assignment/revocation of security attributes to objects take effect immediately.

ACC_RBAC.12 The set of (RBAC) authorized administrative roles is defined as a role that is assigned the following authorizations:

ACC_RBAC.12.1 "hpux.security.access.*"

6.2.2.3. Access Mode Permissions (ACC_MODE_PERS)

ACC_MODE_PERMS.1 Each File System Object is associated with the following attributes:

ACC_MODE_PERMS.1.1 an owning user identification (owner user ID)

ACC_MODE_PERMS.1.2 a group identification (group ID)

ACC_MODE_PERMS.1.3 a set of access permissions

ACC_MODE_PERMS.2 Each System V IPC and POSIX IPC object is associated with the following attributes:

ACC_MODE_PERMS.2.1 an owning user identification (owner user ID)

ACC_MODE_PERMS.2.2 a group identification (group ID)

ACC_MODE_PERMS.2.3 (System V only) a creator user identification (creator user ID)

ACC_MODE_PERMS.2.4 (System V only) a creator group identification (creator group ID)

ACC_MODE_PERMS.2.5 a set of access permissions

ACC_MODE_PERMS.3 The set of access permissions associated with a File System Object shall specify the allowable access modes of the following three classes of (mutually independent) users:

ACC_MODE_PERMS.3.1 The *owner* of the object, identified by the owner user ID associated with the object

ACC_MODE_PERMS.3.2 Any member of the *group* (identified by the group ID) associated with the object (except the owner)

ACC_MODE_PERMS.3.3 Any *other* user (except the owner of the object or any member of the group associated with the object).

ACC_MODE_PERMS.4 The set of access permissions associated with a System V IPC or POSIX IPC object shall specify the allowable access modes of the following three classes of (mutually independent) users:

ACC_MODE_PERMS.4.1 The *owner* and the (System V only) *creator* of the object, identified respectively by the user ID and (System V only) creator user ID associated with the object.

ACC_MODE_PERMS.4.2 Any member of the *group* (identified by the group ID) and (System V only) *creator group* (identified by the creator group ID) associated with the object (except the owner or (System V only) creator)

ACC_MODE_PERMS.4.3 Any *other* user (except the owner or (System V only) creator group associated with the object)

ACC_MODE_PERMS.5 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 object, independently for each class of user (owner, group, other)

Table 6-3 Access Mode Permissions

Access Mode Permissions					
File System Objects			**System V IPC and POSIX IPC Objects**		
Files	**Directories**	**Special Files and Named Pipes**	**Message Queue**	**Shared Memory**	**Semaphore**
Read	Read	Read	Receive	Attach for Read	Read
Write	Write	Write	Send	Attach for Write	Alter
Execute	Search	-	-	-	-

ACC_MODE_PERMS.6 When an unprivileged process requests access to a System V IPC and POSIX IPC object, or makes request to open a File System Object, the access mode permissions for that object shall be checked by the TOE, against the process effective user ID, effective group ID, and any group ID in the process' group access list, to determine whether the process can access the object in the requested mode. (The access check algorithm for File System Objects is specified in ACC_MODE_PERMS.7 and for System V IPC and POSIX IPC objects in ACC_MODE_PERMS.8)

ACC_MODE_PERMS.7 Read, write and execute/search 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:

ACC_MODE_PERMS.7.1 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.

ACC_MODE_PERMS.7.2 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.

ACC_MODE_PERMS.7.3 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.

ACC_MODE_PERMS.7.4 The process has authorized administrator status.

ACC_MODE_PERMS.8 'Receive/(attach for read)/read' and 'send/(attach for write)alter' access to System V IPC and POSIX 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:

ACC_MODE_PERMS.8.1 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.

ACC_MODE_PERMS.8.2 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.

ACC_MODE_PERMS.8.3 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.

ACC_MODE_PERMS.8.4 The process has authorized administrator status.

ACC_MODE_PERMS.9 When a process creates a new File System Object, the object's owner user ID is set to the effective user ID of the process.

ACC_MODE_PERMS.10 When a process creates a new File System Object, the object's group ID is set:

ACC_MODE_PERMS.10.1 to 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.

ACC_MODE_PERS.10.2 to the effective group ID of the process, if the set-group-ID is not present in the parent directory's set of file protection attributes.

ACC_MODE_PERMS.11 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 the process's file mode creation mask (*umask*).

ACC_MODE_PERMS.12 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:

ACC_MODE_PERMS.12.1 The process has ownership rights to the object.

ACC_MODE_PERMS.12.2 The process is privileged, having authorized administrator status.

ACC_MODE_PERMS.13 A process shall be able to change the user and group ownership of a File System Object, provided one or more of the following hold:

ACC_MODE_PERMS.13.1 The process has ownership rights to the object and the process is a member of a privilege group allowing CHOWN.

ACC_MODE_PERMS.13.2 The process is privileged, having authorized administrator status.

ACC_MODE_PERMS.14 When a process creates a new System V IPC and POSIX IPC object, the object's owner user ID and (System V only) creator user ID shall be set to the effective user ID of the process.

ACC_MODE_PERMS.15 When a process creates a new System V IPC or POSIX IPC object, the object's group ID and (System V only) creator group ID shall be set to the effective group ID of the process.

ACC_MODE_PERMS.16 A process shall be able to modify the access mode permissions associated with a System V IPC or POSIX IPC object, provided one or more of the following hold:

ACC_MODE_PERMS.16.1 The process has ownership rights, or (System V only) creator rights, or both ownership and (System V only) creator rights to the object.

ACC_MODE_PERMS.16.2 The process is privileged, having authorized administrator status.

ACC_MODE_PERMS.17 A process shall be able to change the user and group ownership of a System V IPC or POSIX IPC object, provided one or more of the following hold:

ACC_MODE_PERMS.17.1 The process has ownership rights, or (System V only) creator rights, or both ownership and (System V only) creator rights to the object.

ACC_MODE_PERMS.17.2 The process is privileged, having authorized administrator status.

6.2.2.4. HFS Access Control Lists (HFS_ACL)

HFS_ACL.1 Each HFS ACL entry shall specify for one user ID/group combination, a set of access permissions (as specified in the table for ACC_MODE_PERMS.5) to the associated object, which may be zero or more of the following:

HFS_ACL.1	read
HFS_ACL.2	write
HFS_ACL.3	execute/search

HFS_ACL.2 Whenever an unprivileged process makes a request to open an HFS Object, the ACL for that object shall be checked by the TOE's access check algorithm (HFS_ACL.3 and HFS_ACL.4) to determine whether the process can access the object in the requested mode.

HFS_ACL.3 The TOE's access check algorithm checks ACL entries in an object's ACL against the process effective user ID, 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, in the following order of preference:

HFS_ACL.3.1	Specific user, specific group
HFS_ACL.3.2	Specific user, no specific group
HFS_ACL.3.3	No specific user, specific group
HFS_ACL.3.4	No specific user, no specific group

HFS_ACL.4 Where a process has more than one group ID, the TOE's access check algorithm shall set the access mode to the union of the permissions in all matching entries of the same level of precedence.

HFS_ACL.5 A process shall be able to modify the ACL associated with an object, provided one or both of the following hold:

HFS_ACL.5.1	The process has ownership rights to the object
HFS_ACL.5.2	The process has authorized administrator status

HFS_ACL.6 When a process creates a new object, the TOE creates three base ACL entries to correspond with the object's access mode permissions (as determined by ACC_MODE_PERMS.11) as follows:

HFS_ACL.6.1	base ACL entry for the object' *owner* class of user.
HFS_ACL.6.2	base ACL entry for the object's *group* class of user.
HFS_ACL.6.3	base entry for the object's *other* class of user.

HFS_ACL.7 The TOE shall ensure that, irrespective of changes made by users to an object's access mode permissions or ACLs, the 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.

6.2.2.5. VxFS Access Control Lists (VXFS_ACL)

VXFS_ACL.1 Each VxFS ACL (non-default) entry shall specify for one of *owner*, *group,* additional user ID, additional group ID, *other* or group *class*, a set of access permissions (as specified in the table for ACC_MODE_PERMS.5) to the associated object, which may be zero or more of the following:

VXFS_ACL.1.1 read
VXFS_ACL.1.2 write
VXFS_ACL.1.3 execute/search

VXFS_ACL.2 Whenever an unprivileged process makes a request to open a VxFS Object, the ACL for that object shall be checked by the TOE's access check algorithm (VXFS_ACL.3) to determine whether the process can access the object in the requested mode.

VXFS_ACL.3 The TOE's access check algorithm checks ACL entries in an 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 or precedence:

VXFS_ACL.3.1 permissions as specified in the *user* entry

VXFS_ACL.3.2 permissions as specified in the additional user entry, bitwise-AND'd with those in the *class* entry

VXFS_ACL.3.3 Permissions as specified in the *group* entry

VXFS_ACL.3.4 Permissions as specified in the additional group entry, bitwise-AND'd with those in the *class* entry

VXFS_ACL.3.5 Permissions as specified in the *other* entry

VXFS_ACL.4 A process shall be able to modify the ACL associated with an object, provided one or both of the following hold:

VXFS_ACL.4.1 the process has ownership rights to the object
VXFS_ACL.4.2 the process has authorized administrator status

VXFS_ACL.5 When a process creates a new object, the TOE creates four base ACL entries to correspond with the object's access mode permissions (as determined by ACC_MODE_PERMS.11) as follows:

VXFS_ACL.5.1 base ACL entry for the object's *owner* class of user
VXFS_ACL.5.2 base ACL entry for the object's *group* class of user
VXFS_ACL.5.3 base ACL entry for the object's group *class*
VXFS_ACL.5.4 base entry for the object's *other* class of user

VXFS_ACL.6 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.

VXFS_ACL.7 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.

6.2.2.6. 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:

PROC_CTRL.1.1 the real user ID
PROC_CTRL.1.2 the real group ID
PROC_CTRL.1.3 the effective user ID
PROC_CTRL.1.4 the effective group ID
PROC_CTRL.1.5 the group access list
PROC_CTRL.1.6 the process's current working directory
PROC_CTRL.1.7 the audit tag

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:

PROC_CTRL.2.1 the real and effective user IDs are set equal to the user's user ID.

PROC_CTRL.2.2 the real and effective group IDs are set equal to the user's group ID.

PROC_CTRL.2.3 the group access list is set to the set of supplementary group IDs.

PROC_CTRL.2.4 the audit tag is set equal to the audit tag of the parent process (process associated with the user authenticated during the login).

PROC_CTRL.2.5 the current working directory is set equal to the user's home directory.

PROC_CTRL.3 Whenever an executable object is executed by a process, the TOE shall ensure that:

PROC_CTRL.3.1 The 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.

PROC_CTRL.3.2 The 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.

PROC_CTRL.4 Only an authorized administrator or privileged process shall be able to change the real and effective user IDs of a process without re-authentication.

6.2.2.7. Access Policy Enforcement (ACC_POLICY_ENFR)

ACC_POLICY_ENFR.1 The TOE shall validate all attempted operations between subjects and objects, ensuring that all relevant DAC policy enforcement checks succeed before access is granted.

6.2.2.8. Object Reuse (OBJ_REUSE)

OBJ_REUSE.1 The TOE shall ensure that all objects (or parts of objects) are treated before they are assigned to a new subject, such that no conclusion can be drawn regarding the preceding content. The available object reuse resources consist of memory pages, file system objects (FSOs), System V IPC and POSIX IPC objects and Memory Mapped Files (MMFs).

6.2.3. Identification and Authentication (IA)

6.2.3.1. Identification and Authentication Attributes (I&A_ATTR)

I&A_ATTR.1 The TOE shall store the following identification and authentication attributes for each authorized user of the TOE:

- **I&A_ATTR.1.1** user name
- **I&A_ATTR.1.2** user ID
- **I&A_ATTR.1.3** group ID
- **I&A_ATTR.1.4** set of supplementary group IDs (optional)
- **I&A_ATTR.1.6** audit flag
- **I&A_ATTR.1.7** home directory
- **I&A_ATTR.1.8** login program path name
- **I&A_ATTR.1.9** boot flag
- **I&A_ATTR.1.10** encrypted password
- **I&A_ATTR.1.11** password minimum length
- **I&A_ATTR.1.12** whether triviality check is performed on user-generated password
- **I&A_ATTR.1.13** number of unsuccessful login attempts
- **I&A_ATTR.1.14** maximum number of unsuccessful login attempts before the account is locked
- **I&A_ATTR.1.15** account lock flag

I&A_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 own encrypted password entry (I&A_ATTR.1.10) through the trusted interface. Any modification to the user attributes (such as revocation of security attributes) will take place on the next login of the user.

I&A_ATTR.3 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.

I&A_ATTR.4 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 (using roleadm(1m) and authadm(1m) commands):

- **I&A_ATTR.4.1** User passwords
- **I&A_ATTR.4.2** Role definitions and role attributes

6.2.3.2. User Authentication (USR_AUTH)

USR_AUTH.1 The TOE shall authenticate a user's identity before the user is permitted to gain access to the TOE's resources.

USR_AUTH.2 Successful authentication of a user shall require all of the following to be true:

USR_AUTH.2.1 the user name entered by the user exits

USR_AUTH.2.2 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.

USR_AUTH.2.3 except for the root user account at the system console, the user account is not locked.

USR_AUTH.3 The user account shall be locked if any of the following conditions are satisfied:

USR_AUTH.3.1 the user account has been explicitly locked by an authorized administrator.

USR_AUTH.3.2 the number of consecutive unsuccessful attempts to login to the user account exceeds the maximum allowed.

6.2.3.3. Boot Authentication (BOOT_AUTH)

BOOT_AUTH.1 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.

6.2.3.4. User Identification (USR_ID)

USR_ID.1 The TOE shall uniquely identify a user by the user ID associated with that user's user name.

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

6.2.3.5. Password Selection and Generation (PW_SEL&GEN)

PW_SEL&GEN.1 The TOE shall allow users to create user-generated passwords:

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

PW_SEL&GEN.2 User-generated passwords shall comply with the following password construction criteria:

PW_SEL&GEN.2.1 Each password shall have at least six characters. Characters beyond the first eight are ignored.

PW_SEL&GEN.2.2 Each password shall contain at least two alphabetic characters and at least one numeric or special character.

PW_SEL&GEN.2.3 Each password shall differ from the user's user name, and any reverse or circular shift of that user name.

PW_SEL_GEN.2.4 New passwords shall differ from the old password by at least three characters.

6.2.3.6. Password Encryption (PW_ENCR)

PW_ENCR.1 The TOE shall one way encrypt passwords immediately after entry by a user.

PW_ENCR.2 The TOE shall not display passwords in clear text during entry nor store user passwords in clear text.

6.2.4. Protection of TOE Security (PROT)

6.2.4.1. Protection Functions (PROT_FUNCS)

PROT_FUNCS.1 The TOE shall maintain control and data separation between TSF functions executing in kernel space and functions executing in user space.

PROT_FUNCS.2 The TOE shall maintain control and data separation between processes executing in user space.

PROT_FUNCS.3 The TOE shall allow an authorized administrator to run a test utility to confirm that:

PROT_FUNCS.3.1 A user process cannot read or write to system vectors or unmapped areas of virtual memory and that a user process cannot write to read-only areas of virtual memory.

PROT_FUNCS.3.2 The TSF is functioning correctly.

PROT_FUNCS.3.3 The TSF executable code's integrity is verified (e.g., through cksum methods).

PROT_FUNCS.4 If the TOE RBAC mechanism is activated, rbackdbchk(1M) is used in a suite of tests to validate the integrity of the underlying RBAC databases.

PROT_FUNCS.5 If the TOE RBAC mechanism is activated, when any of the RBAC databases (see Roles Based Access Control (RBAC) Mechanism) are off-line, corrupt, or inaccessible all RBAC specific commands (and the associated security functions) will cease functioning.

6.3. Assurance Measures

The assurance measures adopted to satisfy each of the EAL4 assurance requirements, as defined in [CC-V.2.3] Part 3, Section 11.6, Table 10, are summarized in Table 6-4. Note that the assurance components in class ASE defined for EAL4 are met by this document [ST]. See APPENDIX for references.

Table 6-4 Assurance Measures

Assurance Components	Assurance Measures
ACM_AUT.1	This requirement is met by [LCM].
ACM_CAP.4	This requirement is met by [LCM] and [CONFIG].
ACM_SCP.2	This requirement is met by [LCM] and [CONFIG].
ADO_DEL.2	This requirement is met by [LCM].
ADO_IGS.1	This requirement is met by [README], [REL], [INSTALL], [SDAG], [MSW] and [USING]
ADV_FSP.2	This requirement is met by [HLD], which references relevant [MAN PAGES]
ADV_HLD.2	This requirement is met by [HLD], which references relevant [MAN PAGES].
ADV_IMP.1	This requirement is met by HP-UX 11i v3 source code.
ADV_LLD.1	This requirement is met by [LLD].
ADV_RCR.1	This requirement is met by [ST], [HLD], [LLD], and HP-UX 11i v3 source code.
ADV_SPM.1	This requirement is met by this document.
AGD_ADM.1	This requirement is met by [MSW], [ECG], [INSTALL], [README], [REL], [SDAG], and [MAN PAGES].
AGD_USR.1	This requirement is met by [USING], [ECG], [MAN PAGES], and [LCM]
ALC_DVS.1	This requirement is met by [LCM].
ALC_FLR.3	This requirement is met by [LCM] and [ECG]
ALC_LCD.1	This requirement is met by [LCM].
ALC_TAT.1	This requirement is met by [LCM].
ATE_COV.2	This requirement is met by [TPLAN], [TPROC], [STJ], and [STR].
ATE_DPT.1	This requirement is met by [TPLAN] and [HLD].
ATE_FUN.1	This requirement is met by [TPLAN], [TPROC], [STJ], [STR] and [MPR].
ATE_IND.2	Representative platform(s) are provided to enable the evaluators to perform independent functional testing.
AVA_MSU.2	This requirement is met by [VA].
AVA_SOF.1	This requirement is met by [VA].
AVA_VLA.2	This requirement is met by [VA] and [ECG]. Representative platform(s) are provided to enable the evaluators to perform vulnerability testing.

7. PP Claims

7.1. PP Reference

The TOE is in conformance with the Controlled Access Protection Profile [CAPP] and the Role Based Access Control [RBAC] Protection Profile.

7.2. PP Tailoring

TOE security functional requirements are derived from [CAPP] and [RBAC]. They have been tailored by performing operations required by [CAPP] and [RBAC] as defined in Section 5.

Assignments and selections performed by [CAPP] and [RBAC] are highlighted with *italic* fonts. Assignments and selections performed by the Security Target are highlighted with ***bold italic*** fonts. Refinements performed by the Security Target are highlighted with <u>***underlined bold italic***</u> fonts.

7.3. PP Additions

There is one additional SFR in this ST, demanded by the changes to the CC since the PPs were written. FMT_SMF.1 has been added to meet the new dependency requirements of FMT_MSA.1 and FMT_MTD.1.

8. Rationale

This chapter provides the rationale for the selection, creation, and use of the security policies, objectives, and components. Section 8.1 provides the rationale for the existence of the security objectives based upon the stated security assumptions and policies while Section 8.2 provides the low-level rationale for the existence of functional and assurance components based upon the stated security objectives. Section 8.3 provides an analysis that maps security functional requirements to security functions. In providing a mapping for the components and objectives, assurance is gained that the objectives were entirely met. This is further detailed in Table 5-1 and Table 8-1.

8.1. Security Objectives Rationale

The description of security objectives for the TOE and its environment in chapter 4, plus the description of the TOE security environment in chapter 3 are fully compliant with [CAPP] and [RABC]. The security objectives rationale presented in [CAPP] section 7.2.2, with the addition of the following tables satisfies the objectives rationale.

Table 8-1 Threats and Policies to Objectives

Objectives **Threats and Policies**	**O.ACCOUNT**	**O.ADMIN**	**O.AUDIT/ING**	**O.AUTHORIZATION**	**O.DISCRETIONARY_ACCESS**	**O.DUTY**	**O.ENFORCEMENT**	**O.ENTRY**	**O.HIERARCHICAL**	**O.KNOWN**	**O.MANAGE**	**O.RESIDUAL_INFORMATION**	**O.ROLE**
T.ACCESS	*	*	*	*	*		*	*			*	*	*
T.ENTRY			*	*	*	*		*	*				
T.OPERATE				*							*		
T.ROLEDEV				*					*	*			*
P.ACCESS	*	*						*					
P.ACCOUNTABILITY	*		*				*				*		
P.AUTHORIZED_USER				*			*				*		
P.NEED_TO_KNOW					*						*	*	

8.1.1. Complete Coverage – Environmental (Non-IT) Assumptions

This section provides evidence demonstrating coverage of the Non-IT security objectives by the environmental assumptions. The following table shows this assumption to objective mapping.

Table 8-2 Non-IT Security Objectives to Environment Assumptions

Non-IT Security Objectives	Environmental Assumptions
O.CONNECT (RBAC)	A.CONNECT A.ACCESS
O.INSTALL (RBAC & CAPP)	A.MANAGE A.NO_EVIL_ADM A.PEER A.OWNER
O.PHYSICAL (RBAC & CAPP)	A.LOCATE A.PROTECT A.CONNECT A.ASSET
O.CREDEN (CAPP)	A.COOP

8.1.2. Complete Coverage – Threats

The CAPP TOE security objectives have been derived exclusively from statements of organizational security policy, and therefore, there are no explicitly defined CAPP threats countered by this ST.

Table 8-1 shows the Threats defined in the [RBAC], with the T.OPERATE and T.ROLEDEV handled by the inclusion of the [CAPP] and its inclusion in HP-UX 11i v3.

8.1.3. Complete Coverage - Policy

This section provides evidence demonstrating coverage of the Organizational Security Policy by both the IT and Non-IT security objectives. The Table 8-1 shows this objective to policy mapping, and the tables following discuss the coverage for each Security Policy.

The following discussion provides detailed evidence of coverage for each statement of organizational security policy:

P.ACCESS Access rights to specific data object are determined by the owner of the object, the role of the subject attempting access, and the implicit and explicit access rights to the object granted to the role of object owner [RBAC].

This policy is implemented by O.ACCOUNT; O.ADMIN controls the access rights, and O.ENTRY

P.AUTHORIZED_USERS Only those users who have been authorized to access the information within the system may access the system.

This policy is implemented by the O.AUTHORIZATIO objective. The O.MANAGE supports this policy by requiring authorized administrators to be able to manage the functions and O.ENFORECEMENT ensures that functions are invoked and operate correctly.

P.NEED_TO_KNOW The system must limit the access to, modification of, and destruction of the information in protected resources to those authorized users who have a "need to know" for that information.

This policy is implemented by the O.DISCRETIONARY_ACCESS objective. The O.RESIDUAL_INFORMATION objective ensures that information will not be given to users who do not have a need to know, when resources are reused. The O.MANAGE objective supports this policy by requiring authorized administrator be able to manage the functions and O.ENFORCEMENT ensures that functions are invoked and operate correctly.

P.ACCOUNTABILITY The users of the system shall be held accountable for their actions within the system. This policy is implemented by the O.AUDITING objective by requiring that actions are recorded in an audit trail. The O.MANAGE objective supports this policy by requiring authorized administrator be able to manage functions and O.ENFORCEMENT ensures that functions are invoked and operate correctly.

8.2. Security Requirements Rationale

This ST as a whole provides evidence supporting the combined internal consistency and completeness of the functional components that comprise the ST against the [CAPP] and [RBAC] protection profiles. Although there is no Rationale Section within the [RBAC] PP, the Rationale for [CAPP] plus the additional information provided in this section plus other tables accomplishes the requirements.

8.2.1. Security Functional Requirements Cover Security Objectives

The security functional requirements in this ST are derived directly from [CAPP] and [RBAC], with the security objectives that agree with Chapter 4 identified in Table 5-1 and the following table. Therefore, the rationale for Complete Coverage in [CAPP] section 7.2.2 with the amended table below, satisfy the rationale and is not repeated here.

Table 8-3 Security Objectives to SFRs

Security Objectives	Security Functional Requirements
O.ACCOUNT	FIA_ATD.1 FIA_SOS.1 FIA_UAU.2 FIA_UAU.7 FIA_UID.1 FMT_MSA.2 FMT_MTD.1-1/5
O.ADMIN	FAU_GEN.1 FAU_GEN.2

Security Objectives	Security Functional Requirements
	FAU_SAR.1 FAU_SAR.2 FAU_SAR.3 FAU_SEL.1 FAU_STG.1 FIA_USB.1 FMT_MTD.1-1 FMT_MTD.1-2 FMT_MTD.1-5 FPT_STM.1
O.AUTHORIZATION	FDP_ACC.1 FDP_ACF.1 FIA_ATD.1 FIA_SOS.1 FIA_UAU.1 FIA_UAU.7 FIA_UID.1 FMT_MSA.1-2 FMT_MSA.2 FMT_MSA.3 FMT_MTD.1-5
O.AUDIT	FAU_GEN.1 FAU_GEN.2 FAU_SAR.1 FAU_SAR.2 FAU_SAR.3 FAU_SEL.1 FAU_STG.1 FAU_STG.3 FAU_STG.4 FIA_USB.1 FMT_MTD.1-1 FMT_MTD.1-2 FPT_STM.1
O.DISCRETIONARY_ACCESS	FDP_ACC.1 FDP_ACF.1 FIA_ATD.1 FIA_USB.1 FMT_MSA.1 FMT_MSA.3 FMT_REV.1
O.DUTY	FMT_MSA.2 FMT_MTD.1-5 FMT_SMF.1 FMT_SMR.2
O.ENFORCEMENT	FPT_AMT.1 FPT_RVM.1 FPT_SEP.1
O.ENTRY	FDP_ACC.1 FDP_ACF.1 FPT_AMT.1 FPT_RVM.1 FPT_FLS.1 FPT_SEP.1 FTA_LSA.1

Security Objectives	Security Functional Requirements
	FTA_TSE.1
O.HIERARCHICAL	FMT_MTD.1-5 FMT_SMF.1 FMT_SMR.2
O.KNOWN	FIA_UID.2
O.MANAGE	FAU_SAR.1 FAU_SAR.3 FAU_SEL.1 FAU_STG.1 FAU_STG.3 FAU_STG.4 FMT_MTD.1-1 FMT_MTD.1-2 FMT_MTD.1-3 FMT_MTD.1-4 FMT_REV.1 FMT_SMF.1 FMT_SMR.1
O.RESIDUAL_INFORMATION	FDP_RIP.2-1 FDP_RIP.2-2
O.ROLE	FIA_ATD.1 FIA_UID.2 FIA_USB.1 FMT_MSA.3 FPT_RCV.4

8.2.2. Internal Consistency of Requirements

This section describes the mutual support and internal consistency of the components selected for this ST. These properties are discussed for both functional and assurance components.

The security functional requirements components were selected from pre-defined CC components. The use of component refinement was accomplished in accordance with CC guidelines.

Assignment, selection, and refinement operations were carried out among components using consistent computer security terminology. This helps avoid the ambiguity associated with interpretations of meanings of terms between related components.

Multiple instantiation of identical or hierarchically-related components were used where necessary to clearly state the required functionality that must exist in a TOE conformant with these profiles.

8.2.3. Satisfaction of Dependencies

The security functional requirements of the TOE comply with [CAPP] and [RBAC] with no augmentation except for the addition of the dependency for FMT_SMF.1 which is levied on FMT_MSA.1 and FMT_MTD.1. The dependency satisfaction is shown in the following table.

Table 8-4 Dependencies of Security Functional Requirements

Security Functional Requirement	Dependency Satisfied By
FAU_GEN.1	FAU_STM.1
FAU_GEN.2	FAU_GEN.1, FIA_UID.1
FAU_SAR.1	FAU_GEN.1
FAU_SAR.2	FAU_SAR.1
FAU_SAR.3	FAU_SAR.1
FAU_SEL.1	FAU_GEN.1, FAU_MTD.1
FAU_STG.1	FAU_GEN.1
FAU_STG.3	FAU_STG.1
FAU_STG.4	FAU_STG.1
FDP_ACC.1	FDP_ACF.1
FDP_ACF.1	FDP_ACC.1, FMT_MSA.3
FDP_RIP.2-1	None
FDP_RIP.2-2	None
FIA_ATD.1	None
FIA_SOS.1	None
FIA_UAU.2	FIA_UID.2 (Hierarchical to FIA_UID.1)
FIA_UAU.7	FIA_UAU.1
FIA_UID.2	None
FIA_USB.1	FIA_ATD.1
FMT_MSA.1-1	FDP_ACC.1, FMT_SMF.1, FMT_SMR.1
FMT_MSA.1-2	FDP_ACC.1, FMT_SMF.1, FMT_SMR.1
FMT_MSA.2	ADV_SPM.1, FDP_ACC.1, FMT_MSA.1, FMT_SMR.1
FMT_MSA.3	FMT_MSA.1
FMT_MTD.1-1	FMT_SMF.1, FMT_SMR.1
FMT_MTD.1-2	FMT_SMF.1, FMT_SMR.1
FMT_MTD.1-3	FMT_SMF.1, FMT_SMR.1
FMT_MTD.1-4	FMT_SMF.1, FMT_SMR.1
FMT_MTD.1-5	FMT_SMF.1, FMT_SMR.1
FMT_MTD.3	ADV_SPM.1, FMT_MTD.1
FMT_REV.1-1	FMT_SMR.1
FMT_REV.1-2	FMT_SMR.1
FMT_SMF.1	None
FMT_SMR.2	FIA_UID.2 (Hierarchical to FIA_UID.1)
FPT_AMT.1	None
FPT_FLS.1	ADV_SPM.1
FPT_RCV.1	ADV_SPM.1, AGD_ADM.1
FPT_RCV.4	ADV_SPM.1
FPT_RVM.1	None
FPT_SEP.1	None
FPT_STM.1	None

Security Functional Requirement	Dependency Satisfied By
FPT_TST.1	APT_ATM.1
FTA_LSA.1	None
FTA_TSE.1	None

8.2.4. Rationale for Assurance Level

This Security Target has been developed for a generalized environment with a moderate level of risk to the assets. It is intended that TOEs used in these environments will be generally available, without modification to meet the security needs of the environment. As such, it was determined the Evaluation Assurance Level 4, augmented with ALC_FLR.3 Systematic Flaw Remediation, was the most appropriate.

8.2.5. Rationale for SOF Rating

The strength of function rating of SOF-medium is consistent with the EAL4 requirements. SFR FIA_SOS.1 describes how we meet the SOF by providing a ‘one off’ probability of guessing the password to 1 in 1,000,000.

8.2.6. Rationale for Hierachical Roles

The SFR FMT_MTD.1-5 (c) describes how we meet the hierarchical role objective. The rationale below describes how the TOE meets this requirement.

In HP-UX 11i v3 Role-based Access Control, a user may perform privileged actions based on whether the user is ‘authorized’. Specifically, the enforcement point determines whether a particular action is allowed based on whether the user has the necessary ‘authorization’ where authorization is a {user, operation, object} tuple. So, as an example, in order to run the ‘set_parms_addl_netwrk’ command with increased privilege (e.g. uid 0), the user is required to have the authorization (hpux.network.config.*).

A role in HP-UX 11i v3 RBAC is simply a grouping of authorizations. A user is assigned a set of authorizations indirectly by being assigned to a role. Roles serve no other purpose than to simplify authorization assignment, and have no other intrinsic meaning. No APIs are exposed outside of the RBAC subsystem to allow an application to query a user’s role, only whether a user has particular authorization.

8.3. TOE Summary Specification Rationale

Table 8-5 SFR Elements to SF Mapping Rationale

SFR and Component Name	SFR Element	Security Function	Definition
FAU_GEN.1 Audit Data Generation	FAU_GEN.1.1	AUD_DATA_COLL.1 AUD_DATA_COLL.5 AUD_DATA_COLL.6 AUD_EVENTS.2	Audit Data Collection Audit Events

SFR and Component Name	SFR Element	Security Function	Definition
	FAU_GEN.1.2	AUD_RECS.2 AUD_RECS.4 AUD_RECS.5 AUD_RECS.6	Audit Records
FAU_GEN.2 User Identity Association	FAU_GEN.2.1	AUD_RECS.1 AUD_RECS.2	Audit Records
FAU_SAR.1 Audit Review	FAU_SAR.1.1	AUD_LOG_VWNG.1 AUD_LOG_VWNG.2	Audit Logs Viewing
	FAU_SAR.1.2	AUD_LOG_VWNG.1 AUD_LOG_VWNG.2	Audit Logs Viewing
FAU_SAR.2 Restricted Audit Review	FAU_SAR.2.1	AUD_DATA_COLL.4 AUD_LOG_VWNG.1 AUD_LOG_VWNG.2	Audit Data Collection Audit Logs Viewing
FAU_SAR.3 Selectable Audit Review	FAU_SAR.3.1	AUD_LOG_VWNG.1 AUD_LOG_VWNG.2	Audit Logs Viewing
FAU_SEL.1 Selective Audit	FAU_SEL.1.1	AUD_DATA_COLL.3 AUD_EVENTS.1 AUD_EVENTS.3 AUD_EVENTS.4 AUD_EVENTS.5 AUD_LOG_DATA_STRMG.1 AUD_LOG_DATA_STRMG.2 AUD_LOG_DATA_STRMG.3	Audit Data Collection Audit Events Audit Log Data Streamlining
FAU_STG.1 Protected Audit Trail Storage	FAU_STG.1.1	AUD_DATA_COLL.2 AUD_DATA_COLL.4	Audit Data Collection
	FAU_STG.1.2	AUD_DATA_COLL.4 AUD_LOG_FILES_MTNS.8	Audit Data Collection Audit Log Files Maintenance
FAU_STG.3 Action in case of Possible Audit Data Loss	FAU_STG.3.1	AUD_LOG_FILES_MTNS.1 AUD_LOG_FILES_MTNS.2 AUD_LOG_FILES_MTNS.3	Audit Log Files Maintenance
FAU_STG.4 Prevention of Audit Data Loss	FAU_STG.4.1	AUD_LOG_FILES_MTNS.4 AUD_LOG_FILES_MTNS.5 AUD_LOG_FILES_MTNS.6 AUD_LOG_FILES_MTNS.7	Audit Log Files Maintenance
FDP_ACC.1 Subset Access Control	FDP_ACC.1.1	ACC_DAC.1 ACC_RBAC.4 HFS_ACL.7 VXFS_ACL.7	Discretionary Access Control Role Based Access Control HFS Access Control Lists VXFS Access Control Lists
FDP_ACF.1 Security Attribute Based Access Control	FDP_ACF.1.1	ACC_DAC.2 ACC_DAC.3 ACC_RBAC.4 ACC_MODE_PERMS.1 ACC_MODE_PERMS.2 ACC_MODE_PERMS.3 ACC_MODE_PERMS.4 ACC_MODE_PERMS.5 HFS_ACL.1 VXFS_ACL.1	Discretionary Access Control Role Based Access Control Access Mode Permissions HFS Access Control Lists VXFS Access Control Lists

SFR and Component Name	SFR Element	Security Function	Definition
	FDP_ACF.1.2	ACC_RBAC.4 ACC_MODE_PERMS.6 ACC_MODE_PERMS.7 ACC_MODE_PERMS.8 HFS_ACL.2 HFS_ACL.3 HFS_ACL.4 VXFS_ACL.2 VXFS_ACL.3	Role Based Access Control Access Mode Permissions HFS Access Control Lists VXFS Access Control Lists
	FDP_ACF.1.3	ACC_RBAC.4 ACC_MODE_PERMS.7 ACC_MODE_PERMS.8	Role Based Access Control Access Mode Permissions
	FDP_ACF.1.4	ACC_RBAC.4 ACC_MODE_PERMS.6 ACC_MODE_PERMS.7 ACC_MODE_PERMS.8 HFS_ACL.2 HFS_ACL.3 HFS_ACL.4 VXFS_ACL.2 VXFS_ACL.3	Role Based Access Control Access Mode Permissions HFS Access Control Lists VXFS Access Control Lists
FDP_RIP.2-1 Object Residual Information Protection	FDP_RIP.2.1	OBJ_REUSE.1	Object Reuse
FDP_RIP.2-2 Subject Residual Information Protection	FDP_RIP.2.1	OBJ_REUSE.1	Object Reuse
FIA_ATD.1 User Attribute Definition	FIA_ATD.1.1	I&A_ATTR.1 I&A_ATTR.3 ACC_RBAC.1	Identification and Authentication Attributes Role Based Access Control
FIA_SOS.1 Verification of Secrets	FIA_SOS.1.1	USR_AUTH.3 PW_SEL&GEN.1 PW_SEL&GEN.2	User Authentication Password Selection and Generation
FIA_UAU.2 User Authentication Before Any Action	FIA_UAU.2.1	USR_AUTH.1 USR_AUTH.2 USR_AUTH.3 BOOT_AUTH.1	User Authentication Boot Authentication
FIA_UAU.7 Protected Authentication Feedback	FIA_UAU.7.1	PW_ENCR.1 PW_ENCR.2	Password Encryption
FIA_UID.2 User Identification Before Any Action	FIA_UID.2.1	USR_AUTH.1 USR_AUTH.2 BOOT_AUTH.1 USR_ID.1	User Authentication Boot Authentication User Identification
FIA_USB.1 User-Subject Binding	FIA_USB.1.1-1	PROC_CTRL.1	Process Control
	FIA_USB.1.1-2	USR_ID.1 USR_ID.2 PROC_CTRL.2	User Identification Process Control

SFR and Component Name	SFR Element	Security Function	Definition
	FIA_USB.1.1-3	PROC_CTRL.3 PROC_CTRL.4	Process Control
FMT_MSA.1-1 Management of Object Security Attributes	FMT_MSA.1.1	ACC_RBAC.8 ACC_MODE_PERMS.12 ACC_MODE_PERMS.13 ACC_MODE_PERMS.16 ACC_MODE_PERMS.17 HFS_ACL.5 VXFS_ACL.4	Role Based Access Control Access Mode Permissions HFS Access Control Lists VXFS Access Control Lists
FMT_MSA.1-2 Management of Role Security Attributes	FMT_MSA.1.1	ACC_RBAC.5 ACC_RBAC.6	Role Based Access Control
FMT_MSA.2 Secure Security Attributes	FMT_MSA.2.1	ACC_RBAC.7	Role Based Access Control
FMT_MSA.3 Static Attribute Initialization	FMT_MSA.3.1	ACC_RBAC.2 ACC_MODE_PERMS.9 ACC_MODE_PERMS.10 ACC_MODE_PERMS.11 ACC_MODE_PERMS.14 ACC_MODE_PERMS.15 HFS_ACL.6 VXFS_ACL.5 VXFS_ACL.6	Role Based Access Control Access Mode Permissions HFS Access Control Lists VXFS Access Control Lists
	FMT_MSA.3.2	ACC_RBAC.8 ACC_MODE_PERMS.12 ACC_MODE_PERMS.13 ACC_MODE_PERMS.16 ACC_MODE_PERMS.17 HFS_ACL.5 VXFS_ACL.4	Role Based Access Control Access Mode Permissions HFS Access Control Lists VXFS Access Control Lists
FMT_MTD.1-1 Management of Audit Trail	FMT_MTD.1.1	AUD_DATA_COLL.2 AUD_DATA_COLL.3 AUD_DATA_COLL.4 AUD_LOG_FILES_MTNS.1	Audit Data Collection Audit Log Files Maintenance
FMT_MTD.1-2 Management of Audited Events	FMT_MTD.1.1	AUD_DATA_COLL.3 AUD_DATA_COLL.4 AUD_EVENTS.3 AUD_EVENTS.4 AUD_LOG_DATA_STRMG.1 AUD_LOG_VWNG.1	Audit Data Collection Audit Events Audit Log Data Streamlining Audit Logs Viewing
FMT_MTD.1-3 Management of User Attributes	FMT_MTD.1.1	I&A_ATTR.2	Identification and Authentication Attributes
FMT_MTD.1-4 Management of Authentication Data	FMT_MTD.1.1-1	I&A_ATTR.2	Identification and Authentication Attributes
	FMT_MTD.1.1-2	I&A_ATTR.2 PW_ENCR.1 PW_ENCR.2	Identification and Authentication Attributes Password Encryption

SFR and Component Name	SFR Element	Security Function	Definition
FMT_MTD.1-5 Management of TSF Data	FMT_MTD.1.1	ACC_RBAC.1 ACC_RBAC.5 ACC_RBAC.6 ACC_RBAC.9 I&A_ATTR.2 I&A_ATTR.3 I&A_ATTR.4	Role Based Access Control Identification and Authentication Attributes
FMT_MTD.3 Secure TSF Data	FMT_MTD.3.1	ACC_RBAC.7	Role Based Access Control
FMT_REV.1-1 Revocation of User Attributes	FMT_REV.1.1	I&A_ATTR.2 ACC_RBAC.1 ACC_RBAC.5	Identification and Authentication Attributes Role Based Access Control
	FMT_REV.1.2	I&A_ATTR.2 ACC_RBAC.3	Identification and Authentication Attributes Role Based Access Control
FMT_REV.1-2 Revocation of Object Attributes	FMT_REV.1.1	ACC_RBAC.8 ACC_MODE_PERMS.12 HFS_ACL.5 VXFS_ACL.4	Role Based Access Control Access Mode Permissions HFS Access Control Lists VXFS Access Control Lists
	FMT_REV.1.2	ACC_RBAC.10 ACC_MODE_PERMS.6 ACC_MODE_PERMS.7 ACC_MODE_PERMS.8 HFS_ACL.2 HFS_ACL.3 HFS_ACL.4 VXFS_ACL.2 VXFS_ACL.3	Role Based Access Control Access Mode Permissions HFS Access Control Lists VXFS Access Control Lists
FMT_SMF.1 Specification of Management Functions	FMT_SMF.1.1	AUD_LOG_DATA_STRMG.2	Audit Log Data Streamlining
FMT_SMR.2 Security Roles and Restriction on Security Roles	FMT_SMR.1.1	I&A_ATTR.2 PW_SEL&GEN.1 ACC_MODE_PERMS.7 ACC_MODE_PERMS.8 ACC_MODE_PERMS.12 ACC_MODE_PERMS.13 ACC_MODE_PERMS.16 ACC_MODE_PERMS.17 HFS_ACL.5 VXFS_ACL.4 AUD_DATA_COLL.2 AUD_DATA_COLL.3 AUD_DATA_COLL.4 AUD_EVENTS.3 AUD_EVENTS.4 AUD_LOG_DATA_STRMG.1 AUD_LOG_VWNG.1 AUD_LOG_FILES_MTNS.1 AUD_LOG_FILES_MTNS.2 AUD_LOG_FILES_MTNS.3 AUD_LOG_FILES_MTNS.7	Identification and Authentication Attributes Password Selection and Generation Access Mode Permissions HFS Access Control Lists VXFS Access Control Lists Audit Data Collection Audit Events Audit Log Data Streamlining Audit Logs Viewing Audit Log Files Maintenance

SFR and Component Name	SFR Element	Security Function	Definition
	FMT_SMR.1.2	I&A_ATTR.2 ACC_MODE_PERMS.13	Identification and Authentication Attributes Access Mode Permissions
	FMT_SMR.2.1	ACC_RBAC.1 ACC_RBAC.12	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
FPT_AMT.1 Abstract Machine Testing	FPT_AMT.1.1	PROT_FUNCS.3 PROT_FUNCS.4	Protection Functions
FPT_FLS.1 Failure with Preservation of Secure State	FPT_FLS.1.1	PROT_FUNCS.5	Protection Functions
FPT_RCV.1 Manaul Recovery	FPT_RCV.1.1	PROT_FUNCS.5	Protection Functions
FPT_RCV.4 Function Recovery	FPT_RCV.4.1	PROT_FUNCS.5	Protection Functions
FPT_RVM.1 Non-Bypassability of the TSP	FPT_RVM.1.1	ACC_POLICY_ENFR.1	Access Policy Enforcement
FPT_SEP.1 TSF Domain Separation	FPT_SEP.1.1	PROT_FUNCS.1	Protection Functions
	FPT_SEP.1.2	PROT_FUNCS.2	Protection Functions
FPT_STM.1 Reliable Time Stamps	FPT_STM.1.1	AUD_RECS.3	Audit Records
FPT_TST.1 TSF Testing	FPT_TST.1.1	PROT_FUNCS.3	Protection Functions
	FPT_TST.1.2	PROT_FUNCS.4	Protection Functions
	FPT_TST.1.3	PROT_FUNCS.3	Protection Functions
FTA_LSA.1 Limitation on Scope of Selectable Attributes	FTA_LSA.1.1	ACC_RBAC.6	Role Based Access Control
FTA_TSE.1 TOE Session Establishment	FTA_TSE.1.1	ACC_RBAC.4	Role Based Access Control

8.4. PP Claim Ratioanle

The objectives in this ST are from [CAPP] and [RBAC]. The only change is to use the single Objective AUDIT/ING instead of both AUDIT and AUDITING as they have the same meaning in both PPs.

The SFRs used in this ST are derived from [CAPP] and [RBAC]. The required assignments and selections for each SFR are displayed in Chapter 5.

8.4.1. Rationale for Assumptions

- A.MANAGE (CAPP portion) in section 3.1.2 is identical to A.MANAGE in [CAPP], with the exception that we have additionally assumed that the ‘competent’ individuals are also trustworthy (for consistency with [RBAC] PP). It is obvious that A.MANAGE in [CAPP] is not intended to cover individuals that are not ‘trustworthy’, and hence this does not result in any contradiction.
- A.MANAGE in RBAC is split into two parts. The first (general) sentence is included in A.MANAGE (RBAC portion) in section 3.1.2, as this applies to management of TOE security in general, and in this respect overlaps with A.MANAGE in CAPP. The second part is specific to [RABC] and has been incorporated in A.MANAGE (RBAC portion) in section 3.1.2 as well.

8.4.2. Rationale for FMT_REV.1.1

- The [CAPP] FMT_REV.1 requirements are satisfied because the text used in the corresponding ST SFRs is identical to those in the PP, with one exception: in section 5.1.4.1.2, the FMT_REV.1.1 SFR is refined so that the “Object Owner” and “Authorized Administrator” roles are explicitly identified for consistency with the [RBAC] PP. The “Object Owner” role was only implicit in [CAPP] (for example, in CAPP 5.4.1.1).
- The formatting error in [CAPP] has been corrected. In section 5.1.4.12, FMT_REV from [CAPP] is changed to FMT_REV.1.2.
- [RBAC 5.1.4] is satisfied in the following ways:
    - The “set of RBAC administrative roles” is defined to be the set of “authorized administrators” as specified in [CAPP].
    - FMT_REV.1.2 in 5.1.4.11, rule b) is equivalent to the RBAC requirement that user security attributes are revoked on the next login of the user.
    - FMT_REV.1.2 in 5.1.4.12, rule a) is equivalent to the RBAC requirement that object security attributes are revoked on the next attempt to access the object.

APPENDIX A: References

[CC-V2.3] Common Criteria for Information Technology Security Evaluation, ISO/IEC 15408, Version 2.3, August 31, 2005:

Part 1 Introduction and general model, CCMB-2005-08-001
Part 2 Security functional requirements, CCMB-2005-08-002
Part 3 Security assurance requirements, CCMB-2005-08-003

[CAPP] Controlled Access Protection Profile, NSA, Version 1.d, October 8, 1999

[RBAC] Role Based Access Control Protection Profile (RBAC), version 1.0, July 30, 1998

[MPR] Multi-Platform Rationale, HP-UX 11i v3 Common Criteria, Hewlett-Packard, September, 2007

[HLD] High Level Design Document, HP-UX 11i v3 Common Criteria, Hewlett-Packard, December 2007

[LLD] Low-Level Design Document, HP-UX 11i v3 Common Criteria Hewlett-Packard:

Part 1: Instantiation of Security Functions, December 2007
Part 2: Design Specifications of HP-UX Subsystems, December 2007

[TPLAN] Test Plan Document, HP-UX 11i v3 Common Criteria, Hewlett-Packard, December 2007

[TPROC] Test Procedure, HP-UX 11i v3 Common Criteria, Hewlett-Packard, December 2007

[STJ] Test Journal Raw Data (IPF) and Test Journal Raw Data (PA), HP-UX 11i v3 Common Criteria, Hewlett-Packard, December 2007

[STR] Test Report Document, HP-UX 11i v3 Common Criteria, Hewlett-Packard, December 2007

[VA] Vulnerability Assessment, HP-UX 11i v3 Common Criteria, Hewlett-Packard, September 2007

[LCM] Life Cycle Management, HP-UX 11i v3 Common Criteria, Hewlett-Packard, December 2007

[ECG] Common Criteria HP-UX 11i v3 Evaluated Configuration Guide, HP9000 and HP Integrity Computers, Hewlett-Packard, December 2007

[INSTALL] HP-UX 11i v3 Installation and Update Guide: HP Integrity Servers and HP 9000 Servers, Edition 1, Hewlett-Packard, February 2007

[MAN PAGES] HP-UX Reference (Volumes 1 to 9) HP-UX 11i Version 3, Hewlett-Packard, February 2007

[MSW] Managing Systems and Workgroups: A Guide for HP-UX System Administrators, Hewlett-Packard, Version 9, March 2006,

[README] Read Before Installing or Updating HP-UX 11i v3, Hewlett-Packard, December 2007

[REL] HP-UX 11i Version 3 Release Notes, Hewlett-Packard, February 2007,

[SDAG] Software Distributor Administration Guide for HP-UX 11i v3, Hewlett-Packard, 5992-2146, September 2007

[USING] Using HP-UX, Hewlett-Packard, September 1997

APPENDIX B: Acronyms

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