Security Target

HP-UX 11.11 Common Criteria

Prepared by: Ralph Worswick

Project Manager: Chris Simpson
Pinewood
Nine Mile Ride
Wokingham
Berks
RG40 3LL
Telnet No: 316-5029

Project Document Id: HPUX11CC-TR-01
Date Prepared: September 2002

HP-UX 11.11 CC	Security Target	

Document Information

Project Name:	HP-UX 11.11 Common Criteria		
Project Manager:	Chris Simpson	**Document Version No:**	06
FocusPM Phase:		**Document Version Date:**	25/09/2002
Quality Review Method:	Peer Review		
Prepared By:	Ralph Worswick	**Preparation Date:**	25/09/2002
Reviewed By:	CMG, Chris Simpson	**Review Date**	

Distribution List

From	Date	Phone/Fax
Ralph Worswick	25/09/2002	01276 687314

To	Action*	Date	Phone/Fax
Jo O'Donoghue	File	25/09/2002	
CMG CLEF	Inform	25/09/2002	
CMG	Inform	25/09/2002	
Chris Simpson David Garner	Inform Inform	25/09/2002 25/09/2002	
Pete Tompkins	Inform	25/09/2002	
Alan Greer	Inform	25/09/2002	

** Action Types: Approve, Review, Inform, File, Action Required, Attend Meeting, Other (please specify)*

Version History

Ver. No.	Ver. Date	Revised by	Description	Filename
01	09/01	RW	First Draft	Security Target Draft A2
02	10/01	RW	First Issue	Security Target Issue 1.0
03	06/02	RW	Draft Issue 2	Security Target Draft 02
04	07/02	RW	Second Issue	TR01I02
05	09/02	RW	Third Issue	TR01I03
06	09/02	RW	Fourth Issue	TR01I04

Proprietary Notice

Contents

References

References

[ASD]	Architecture Summary Document, HP-UX 11i Common Criteria, Hewlett-Packard, HPUX11CC-TB-01, Issue 1, 24 January 2002
[CAPP]	Controlled Access Protection Profile, NSA, Version 1.d, 8 October 1999
[CC]	Common Criteria for Information Technology Security Evaluation, ISO/IEC 15408, Version 2.1, August 1999:
	Part 1 Introduction and general model, CCIB-99-031
	Part 2 Security functional requirements, CCIB-99-031
	Part 3 Security assurance requirements, CCIB-98-028
[CCNTL]	Configuration Control CD
[ECG]	Common Criteria HP-UX 11i Evaluated Configuration Guide HP9000 Computers, Hewlett-Packard, 5990-3527, May 2002
[FS]	Functional Specification, HP-UX 11.11 Common Criteria, Hewlett-Packard, HPUX11CC-TB-03, Issue 2, July 2002
[HLD]	High Level Design, HP-UX 11i Common Criteria, Hewlett-Packard, HPUX11CC-TB-02, Issue 2, July 2002
[INSTALL]	HP-UX 11i Installation and Update Guide, Hewlett-Packard, 5185-6511, Edition 3, September 2001(http://www.docs.hp.com/hpux/os/11i/index.html)
[ITSEC ST]	ITSEC Security Target HP9000 Computer Systems, Hewlett-Packard, ITSEC-ST-HPUX1020, Version 5.0, 30 July 1998
[LLD]	Low Level Design – Access Control, HP-UX 11.11 Common Criteria, Hewlett-Packard, HPUX11CC-TD-02, Issue 1.0, July 2002
	Low Level Design – Audit, HP-UX 11.11 Common Criteria, Hewlett-Packard, HPUX11CC-TD-03, Issue 1.0, July 2002
	Low Level Design – Identification & Authentication, HP-UX 11.11 Common Criteria, Hewlett-Packard, HPUX11CC-TD-04, Issue 1.0, July 2002
	Low Level Design – New Components, HP-UX 11.11 Common Criteria, Hewlett-Packard, HPUX11CC-TD-05, Issue 1.0, July 2002
	Low Level Design – Object Reuse, HP-UX 11.11 Common Criteria, Hewlett-Packard, HPUX11CC-TD-06, Issue 1.0, July 2002
	Low Level Design – Security Relevant, HP-UX 11.11 Common Criteria, Hewlett-Packard, HPUX11CC-TD-07, Issue 1.0, July 2002
[Man Pages]	HP-UX Reference (Volumes 1 to 9) (http://www.docs.hp.com/hpux/os/11i/index.html)
[MPR]	HP-UX 11i Multi-Platform Rationale, Hewlett-Packard, HPUX11CC-TN-01, Issue 3.0, 12 February 2002
[MSU]	Misuse Analysis, HP-UX 11.11 Common Criteria, Hewlett-Packard, HPUX11CC-TM-01, Issue 2.0, July 2002

[MSW]	Managing Systems and Workgroups: A Guide for HP-UX System Administrators, Hewlett-Packard, B2355-90742 E0601, Edition 5, June 2001.
[README]	Read Before Installing or Updating to HP-UX 11i, Hewlett-Packard, 5185-6518, September 2001 (http://www.docs.hp.com/hpux/os/11i/index.html)
[REL1]	HP-UX 11i Release Notes (http://www.docs.hp.com/hpux/11i/index.html)
[REL2]	HP-UX 11i September 2001 Release Notes, Edition 3. 5185-6522 E0901. (http://www.docs.hp.com/hpux/11i/index.html)
[SDAG]	Software Distributor Administration Guide for HP-UX 11i (http://www.docs.hp.com/hpux/11i/index.html)
[SOF]	Strength of Function Analysis, HP-UX 11.11 Common Criteria, Hewlett-Packard, HPUX11CC-TP-01, Issue 1, 15 May 2002
[TCSEC]	Department of Defense Trusted Computer System Evaluation Criteria (TCSEC), National Computer Security Center, DOD 5200.28-STD, December 1985
[TD]	Configuration Control CD…
[TPLAN]	Test Plan, HP-UX 11.11 Common Criteria EAL4/CAPP Evaluation, Hewlett-Packard, Version
[TPROC]	Test Procedures, HP-UX 11.11 Common Criteria, Version
[STJ]	Security Test Journal
[STR]	Security Test Report
[USING]	Using HP-UX (http://www.docs.hp.com/hpux/11.0/index.html)
[VA]	Vulnerability Analysis, HP-UX 11.11 Common Criteria, Hewlett-Packard, HPUX11CC-TC-02, Issue 2.0, July 2002

1 Introduction

1.1 ST Identification

1.1.1 Title: Security Target for HP-UX 11.11.

1.1.2 Keywords: HP-UX 11i, UNIX, POSIX, general purpose operating system.

1.1.3 This document is the Security Target for Hewlett-Packard's general purpose UNIX operating system HP-UX 11.11, also known as HP-UX 11i version 1.0. The Security Target is conformant to the Common Criteria [CC].

1.2 ST Overview

1.2.1 This Security Target specifies the security environment, objectives and features of Hewlett-Packard's general purpose UNIX operating system HP-UX 11.11 (referred to as 'the product') as submitted for evaluation to the [CC] evaluation assurance level EAL4.

1.2.2 The product was designed to exceed the [TCSEC] Class C2 functionality requirements, notable extensions being access control lists - a [TCSEC] Class B3 feature – and boot authentication. The [TCSEC] Class C2 requirements are described for [CC] in the Controlled Access Protection Profile [CAPP].

1.2.3 The product may execute on a single HP 9000 Server or be connected to other HP 9000 Servers executing identical versions of the product to form a local distributed system implementing a unified security policy.

1.3 CC Conformance

1.3.1 The product is conformant with the Controlled Access Protection Profile [CAPP].

1.3.2 The product is [CC] Part 2 extended, Part 3 conformant with a claimed assurance level of EAL4.

1.4 Document Structure

1.4.1 Section 2 provides the description of the TOE.

1.4.2 Section 3 provides the statement of the TOE security environment.

1.4.3 Section 4 provides the statement of the security objectives.

1.4.4 Section 5 provides the statement of the IT security requirements

1.4.5 Section 6 provides the TOE summary specification.

1.4.6 Section 7 provides the statement of PP claims.

1.4.7 Section 8 provides the rationale.

1.5 Conventions

1.5.1 Security functional requirements specified in Section 5.1, tailored by carrying out the operations required by [CAPP], are presented as labelled paragraphs, where the label is a mnemonic corresponding to the security functional requirement derived from [CAPP]. As described in Paragraph 6.1.2, [CAPP] security functional requirements deviating from [CC] are shown by mnemonics in single quotes; iterated [CC] security functional requirements are shown by mnemonics marked with a numeric superscript.

1.5.2 Security functions specified in Section 6.2 are presented as labelled paragraphs and lists, where the label is an abbreviation followed by a sequence number and, in some instances, sub-sequence number in square brackets (e.g. [AC36.2]). Security function references in the text exclude the square brackets.

1.5.3 Security functions are derived mainly from the security enforcing functions (SEFs) specified in the Security Target [ITSEC ST] for the E3 evaluated version of HP-UX 10.20 and are identical to the SEFs in many cases. Security functions that are additional to, or modifications of, [ITSEC ST] SEFs are presented in underlined text.

1.5.4 Document references are identified as abbreviations in square brackets.

1.6 Terminology

Terms used in this document are as defined in Section 6.1, in [CC] and in [CAPP] Section 1.5, with the following elaborations:

a) The term *user* is generally used to mean an *authorized user*

b) An *authorized administrator* of the TOE is the root user, having an effective user ID of zero, considered to have *superuser* status

c) The TOE implements the following security relevant authorizations:

   i) A user granted superuser status
   ii) A user granted the change ownership (CHOWN) capability, which allows the user to change the ownership of File System Objects currently owned by that user.

2 TOE Description

2.1 Product Type

2.1.1 The product is an 'evaluated configuration' (as defined in Section 2.2) of Release 11.11 of the HP-UX operating system (HP-UX 11.11). The product may execute on a single HP 9000 Server or be connected to other HP 9000 Servers executing identical versions of the product to form a local distributed system. The product is also referred to, in [CC] terminology, as the Target of Evaluation (TOE).

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

2.2 Evaluated Configuration

The evaluated configurations of the product are defined as follows:

a) The product executes on any single 64-bit computer system from the family of HP 9000 Servers, including servers offering hardware partitions (nPartitions), which enable a single computer to run more than one copy of the product

b) The product executes on a single HP 9000 Server, which may be connected to other HP 9000 Servers via a local Ethernet network, each executing the same version of the product. No other processors may be connected to the product, either directly by hardwire connection (e.g. to implement a Cluster of HP 9000 systems) or indirectly by, for example, a Wide Area Network or telephone cable to provide remote computer or network services

c) The product supports virtual partitions (VPARs), which enable a single computer to run more than one copy of the product

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

e) The product supports the HFS and JFS File Systems, but excludes Online JFS

f) The product includes Pluggable Authentication Modules (PAM), with the default configuration for authentication consisting of user identity and password

g) The product executes with HP-VUE and X-Windows disabled and excludes the use of a restricted configuration of the System Administration Manager (Restricted SAM)

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

   i) ftp(1)
   ii) rexec(1)
   iii) rlogin(1)
   iv) telnet(1).

i) The product has been installed, set up, converted to a 'Trusted System', and operated as described in [ECG], [INSTALL], [MSW], [REL1], [REL2], [README], [SDAG], and [USING]

j) Boot authentication is enabled and auditing is enabled in multi-user mode, as described in [ECG].

2.3 Summary of Security Features

2.3.1 Introduction

The main security features of the product are:

a) user identification and authentication

b) discretionary access control (DAC), including access control lists

c) auditing.

2.3.2 Identification and Authentication

2.3.2.1 All users of the product are authenticated and held accountable for their security related actions. Each user is uniquely identified by the product. The product records security related events and the user associated with the event.

2.3.2.2 The product supports an ordinary *user* role and a *superuser* (administrative) role.

2.3.2.3 A superuser has 'root privilege' and is not constrained by the product's security policies.

2.3.2.4 An ordinary user does not have 'root privilege' and is constrained by the product's security policies.

2.3.2.5 The product allows a superuser to associate individual users with a privileged group, thus permitting a process acting on the user's behalf to change the ownership of files.

2.3.2.6 The authentication features are supported by constraints on user-generation of passwords and an encryption mechanism.

2.3.3 Discretionary Access Control

2.3.3.1 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.

2.3.3.2 The product implements a DAC policy, which provides both the traditional UNIX 'owner', 'group', 'other' access mode permissions and a more granular access control list (ACL) mechanism, controlled by the object's owner.

2.3.3.3 The product implements two independent ACL mechanisms:

a) HFS ACLs for the HFS File System; and

b) JFS ACLs for the JFS File System.

2.3.3.4 DAC is supported by object reuse mechanisms to ensure that information is not inadvertently transferred between subjects when objects are re-allocated.

2.3.4 Auditing

2.3.4.1 The product is capable of collecting audit records for all security relevant events that occur. A superuser may select the users and events for which audit data is collected from time to time.

2.3.4.2 Audit records may be viewed by a superuser selectively for any period on the basis of criteria such as user name, event type and outcome.

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

3 TOE Security Environment

3.1 Assumptions

The assumptions are fully conformant with [CAPP].

3.1.2 Physical Assumptions

A.LOCATE

The processing resources of the TOE will be located within controlled access facilities which will prevent unauthorized physical access.

A.PROTECT

The TOE hardware and software critical to security policy enforcement will be protected from unauthorized physical modification.

3.1.3 Personnel Assumptions

A.MANAGE

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

A.NO_EVIL_ADM

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

A.COOP

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

3.1.4 Connectivity Assumptions

A.PEER

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

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.

3.2 Threats

There is no statement of explicit threats countered by the TOE, which is conformant with [CAPP].

3.3 Organisational Security Policies

The organisational security policies are fully conformant with [CAPP].

P.AUTHORIZED_USERS

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

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 which have a “need to know” for that information.

P.ACCOUNTABILITY

The users of the system shall be held accountable for their actions within the system.

4 Security Objectives

4.1 Security Objectives for the TOE

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

O.AUTHORIZATION

The TSF must ensure that only authorized users gain access to the TOE and its resources.

O.DISCRETIONARY_ACCESS

The TSF must control accessed 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

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

The TSF must ensure that any information contained in a protected resource is not released when the resource is recycled.

O.MANAGE

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

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

4.2 Security Objectives for the Environment

The security objectives for the Environment are fully conformant with [CAPP].

O.INSTALL

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

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

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.

5 Security Requirements

5.1 TOE Security Functional Requirements

5.1.1 The security functional requirements for the TOE are listed in Table 5.1. They comprise all of the security functional requirements taken from [CAPP]. Functional elements that have been tailored by performing the operations required by [CAPP] are indicated by a $^{\&}$ superscript. Tailored requirements are defined in this section following Table 5.1, with assignments and selections underlined.

5.1.2 [CAPP] draws its security functional requirements from Part 2 of the [CC], with some deviations (including extensions) applied that are described as 'Notes' in Section 8.0 of [CAPP]. In Table 5.1, requirements deviating from [CC] Part 2 are listed in single quotes. [CAPP] also iterates some of the [CC] Part 2 components and functional elements. Table 5.1 marks iterated components and elements with a numeric superscript.

Table 5.1 Security Functional Requirements

Component	Component Name	Functional Element	[CAPP] Paragraph
FAU_GEN.1	Audit Data Generation	FAU_GEN.1.1 FAU_GEN.1.2	5.1.1.1 5.1.1.2
FAU_GEN.2	User Identity Association	FAU_GEN.2.1	5.1.2.1
FAU_SAR.1	Audit Review	FAU_SAR.1.1 FAU_SAR.1.2	5.1.3.1 5.1.3.2
FAU_SAR.2	Restricted Audit Review	FAU_SAR.2.1	5.1.4.1
FAU_SAR.3	Selectable Audit Review	FAU_SAR.3.1$^{\&}$	5.1.5.1
FAU_SEL.1	Selective Audit	FAU_SEL.1.1$^{\&}$	5.1.6.1
FAU_STG.1	Guarantees of Audit Data Availability	FAU_STG.1.1 FAU_STG.1.2	5.1.7.1 5.1.7.2
FAU_STG.3	Action in Case of Possible Audit Data Loss	FAU_STG.3.1$^{\&}$	5.1.8.1
'FAU_STG.4'	Prevention of Audit Data Loss	'FAU_STG.4.1'$^{\&}$	5.1.9.1
FDP_ACC.1	Discretionary Access Control Policy	FDP_ACC.1.1$^{\&}$	5.2.1.1
FDP_ACF.1	Discretionary Access Control Functions	FDP_ACF.1.1$^{\&}$ FDP_ACF.1.2$^{\&}$ FDP_ACF.1.3$^{\&}$ FDP_ACF.1.4$^{\&}$	5.2.2.1 5.2.2.2 5.2.2.3 5.2.2.4
FDP_RIP.2[1]	Object Residual Information Protection	FDP_RIP.2[1].1	5.2.3.1
'FDP_RIP.2[2]' (Note 1)	Subject Residual Information Protection	'FDP_RIP.2[2].1' (Note 1)	5.2.4.1
FIA_ATD.1	User Attribute Definition	FIA_ATD.1.1$^{\&}$	5.3.1.1

Table 5.1 Security Functional Requirements

Component	Component Name	Functional Element	[CAPP] Paragraph
FIA_SOS.1	Strength of Authentication Data	FIA_SOS.1.1	5.3.2.1
FIA_UAU.1	Authentication	FIA_UAU.1.1[&] FIA_UAU.1.2	5.3.3.1 5.3.3.2
FIA_UAU.7	Protected Authentication Feedback	FIA_UAU.7.1	5.3.4.1
FIA_UID.1	Identification	FIA_UID.1.1[&] FIA_UID.1.2	5.3.5.1 5.3.5.2
'FIA_USB.1' (Note 2)	User-Subject Binding	'FIA_USB.1.1'[&] 'FIA_USB.1.2'[&] 'FIA_USB.1.3'[&] (Note 2)	5.3.6.1 5.3.6.2 5.3.6.3
FMT_MSA.1	Management of Object Security Attributes	FMT_MSA.1.1[&]	5.4.1.1
FMT_MSA.3	Static Attribute Initialisation	FMT_MSA.3.1 FMT_MSA.3.2[&]	5.4.2.1 5.4.2.2
FMT_MTD.1[1]	Management of the Audit Trail	FMT_MTD.1[1].1	5.4.3.1
FMT_MTD.1[2]	Management of Audited Events	FMT_MTD.1[2].1	5.4.4.1
FMT_MTD.1[3]	Management of User Attributes	FMT_MTD.1[3].1	5.4.5.1
FMT_MTD.1[4]	Management of Authentication Data	FMT_MTD.1[4].1[1] FMT_MTD.1[4].1[2]	5.4.6.1 5.4.6.2
FMT_REV.1[1]	Revocation of User Attributes	FMT_REV.1[1].1 FMT_REV.1[1].2[&]	5.4.7.1 5.4.7.2
FMT_REV.1[2]	Revocation of Object Attributes	FMT_REV.1[2].1 FMT_REV.1[2].2[&]	5.4.8 5.4.8.1
FMT_SMR.1	Security Management Roles	FMT_SMR.1.1[&] FMT_SMR.1.2	5.4.9.1 5.4.9.2
FPT_AMT.1	Abstract Machine Testing	FPT_AMT.1.1[&]	5.5.1.1
FPT_RVM.1	Reference Mediation	FPT_RVM.1.1	5.5.2.1
FPT_SEP.1	Domain Separation	FPT_SEP.1.1 FPT_SEP.1.2	5.5.3.1 5.5.3.2
FPT_STM.1	Reliable Time Stamps	FPT_STM.1.1	5.5.4.1

FAU_SAR.3.1 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.

FAU_SEL.1.1 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;

b) Event type;

c) System call.

FAU_STG.3.1 The TSF shall generate an alarm to the authorized administrator if the audit trail exceeds an authorized administrator configurable percentage of the allocated file space.

'FAU_STG.4.1' The TSF shall be able to prevent auditable events, except those taken by the authorized administrator, if the audit trail is full.

FDP_ACC.1.1 The TSF shall enforce the Discretionary Access Control Policy on processes acting on the behalf of users, File System, System V IPC and POSIX IPC objects and all operations among subjects and objects covered by the DAC policy.

FDP_ACF.1.1 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 JFS File System objects, the Access Mode Permissions and the JFS ACL;

iii) For System V IPC and POSIX IPC objects, the Access Mode Permissions.

FDP_ACF.1.2 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:

- 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, specific group ACL entry;
- 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;
- 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;
- Access is granted or denied according to the permissions of the default no specific user, no specific group ACL entry;

ii) 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:
- Access is granted or denied according to the permissions if there is a match with the object's *owner* class of user;
- Access is granted or denied according to the permissions if there is a match with the object's *group* class of user;
- Access is granted or denied according to the permissions of the object's *other* class of user;

b) For JFS File System objects:

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:
- 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;
- 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;
- 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;
- 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;
- 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:
- 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;
- 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;
- Access is granted or denied according to the permissions of the object's *other* class of user.

FDP_ACF.1.3 The TSF shall explicitly authorize access of subjects to objects based on the following additional rule:

a) An authorised administrator acting as superuser (with an effective user identity equal to zero) shall be granted access to all objects, overriding the rules specified in FDP_ACF.1.2.

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on no other rules than those specified in FDP_ACF.1.2.

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

a) User Identifier;

b) Group Memberships;

c) Authentication Data;

d) Security-relevant Roles;

e) Audit identity;

f) Home directory;

g) Login program;

h) Audit flag; and

i) Boot flag.

FIA_UAU.1.1 The TSF shall allow no actions on behalf of the user to be performed before the user is authenticated.

FIA_UID.1.1 The TSF shall allow no actions on behalf of the user to be performed before the user is identified.

'FIA_USB.1.1' The TSF shall associate the following user security attributes with subjects acting on the behalf of that user:

a) The user identity which is associated with auditable events;

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

c) The group membership or memberships used to enforce the Discretionary Access Control Policy;

d) The current working directory.

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

a) The user identity which is associated with auditable events is set to the user's Audit Identity;

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.1.3' 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 authorised administrator acting as superuser (with an effective user identity equal to zero) 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()* command);

b) A subject's effective user identity is changed to the owner of a file executed with its set-user-identity permission bit enabled;

c) A subject's effective group identity is changed to the owning group of a file executed with its set-group-identity permission bit enabled.

FMT_MSA.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 superuser (with an effective user identity equal to zero) may modify any access control attributes.

FMT_MSA.3.2 The TSF shall allow the authorised 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.

FMT_REV.1[1].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.

FMT_REV.1[2].2 The TSF shall enforce the rules:

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

FMT_SMR.1.1 The TSF shall maintain the roles:

a) authorized administrator;

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

c) users authorized to modify their own authentication data.

FPT_AMT.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.

5.2 Strength of Functions

The claimed minimum strength of function is SOF-Medium.

5.3 TOE Security Assurance Requirements

The TOE security assurance requirements are those of evaluation assurance level EAL4 with no augmentation or extension.

5.4 Security Requirements for the IT Environment

There are no security requirements for the IT environment.

6 TOE Summary Specification

6.1 Concepts and Terminology

6.1.1 Subjects, Sessions and Privileges

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

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

6.1.1.3 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's real and effective user IDs

e) the process's real and effective group IDs

f) a group access list

g) an audit ID

h) the current working directory.

6.1.1.4 A user gains initial access to the product 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 processes.

6.1.1.5 The first process created in a session is known as the 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 process's parent process ID is the process ID of its parent process.

6.1.1.6 The other security relevant attributes (Paragraph 6.1.1.3 (d) to (h)) 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 ID is set equal to the user's audit ID

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

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

6.1.1.8 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 security relevant attributes of Paragraph 6.1.1.6 that are associated with their parent process.

6.1.1.9 Whenever a process executes an executable object, the effective user and group IDs may be changed. However, the audit ID will not be changed, thus maintaining user accountability for actions.

6.1.1.10 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 ID will not be changed, thus maintaining the initial user's accountability for actions.

6.1.1.11 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.

6.1.1.12 The product provides the following types of privilege:

a) superuser 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.

6.1.1.13 A process with superuser status is not constrained by the product's security policies.

6.1.1.14 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.

6.1.2 Objects and Access Permissions

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

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

6.1.2.3 The product 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 and POSIX IPC (Inter-Process Communication) objects, as follows:

i) message queues
ii) shared memory
iii) semaphore.

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

6.1.2.5 The product implements two 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 objects

b) an Access Control List (ACL) mechanism which, for File System Objects only, further qualifies the access given by the access mode permissions.

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

6.1.3 Initial and Secure States

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

6.1.3.2 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 product implements boot authentication, whereby even root (or privileged) users accessing the product in single user state are authenticated.

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

6.1.4 Security Policy Rationales

6.1.4.1 The product 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.

6.1.4.2 The intent of the DAC policy is twofold:

a) to allow users control over access to objects under their management

b) to protect user activities from undesired interference.

6.2 TOE Security Functions

6.2.1 Identification and Authentication

Identification and Authentication Attributes

6.2.1.1 [I&A1] The product shall store the following identification and authentication attributes for each authorised user of the product:

a) [I&A1.1] user name

b) [I&A1.2] user ID

c) [I&A1.3] group ID

d) [I&A1.4] set of supplementary group IDs (optional)

e) [I&A1.5] audit ID

f) [I&A1.6] audit flag

g) [I&A1.7] home directory

h) [I&A1.8] login program path name

i) [I&A1.9] boot flag

j) [I&A1.10] encrypted password

k) [I&A1.11] Removed

l) [I&A1.12] password maximum length

m) [I&A1.13] Removed

n) [I&A1.14] Removed

o) [I&A1.15] Removed

p) [I&A1.16] Removed

q) [I&A1.17] Removed

r) [I&A1.18] Removed

s) [I&A1.19] Removed

t) [I&A1.20] whether triviality check is performed on user-generated password

u) [I&A1.21] Removed

v) [I&A1.22] Removed

w) [I&A1.23] Removed

x) [I&A1.24] Removed

y) [I&A1.25] Removed

z) [I&A1.26] Removed

aa) [I&A1.27] number of unsuccessful login attempts

bb) [I&A1.28] maximum number of unsuccessful login attempts before the account is locked

cc) [I&A1.29] account lock flag.

6.2.1.2 [I&A5] The product 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 root user can modify the identification and authentication attributes. Non-root users shall be able to modify their own encrypted password entry (I&A1.10) through a trusted interface.

User Authentication

6.2.1.3 [I&A6] The product shall authenticate a user's identity before the user is permitted to gain access to the product's resources.

6.2.1.4 [I&A7] Successful authentication of a user shall require all of the following to be true:

a) [I&A7.1] the user name entered by the user exists

b) [I&A7.2] the password entered by the user, and one way encrypted by the product, is identical to the encrypted password stored by the product for the entered user name

c) [I&A7.3] except for the root user account at the system console, the user account is not locked.

6.2.1.5 [I&A8] The user account shall be locked if any of the following conditions are satisfied:

a) [I&A8.1] the user account has been explicitly locked by a superuser

b) [I&A8.2] the number of consecutive unsuccessful attempts to login to the user account exceeds the maximum allowed.

Boot Authentication

6.2.1.6 [I&A11] The product 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 product into single-user mode.

User Identification

6.2.1.7 [I&A13] The product shall uniquely identify a user by the user ID associated with that user's user name.

6.2.1.8 [I&A14] The product shall enforce individual accountability by associating the audit ID, associated with a user's user name, with all auditable actions performed by the product on behalf of that user.

Password Selection and Generation

6.2.1.9 [I&A16] The product shall allow users to create user-generated passwords.

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

6.2.1.11 [I&A18] User-generated passwords shall comply with the following password construction criteria:

a) [I&A18.1] each password shall have at least six characters and no more than eighty

b) [I&A18.2] each password shall contain at least two alphabetic characters and at least one numeric or special character

c) [I&A18.3] each password shall differ from the user's user name, and any reverse or circular shift of that user name

d) [I&A18.4] new passwords shall differ from the old password by at least one character.

Password Encryption

6.2.1.12 [I&A23] The product shall one way encrypt passwords immediately after entry by a user.

6.2.1.13 [I&A24] The product shall not display passwords in clear text during entry or store user passwords in clear text.

6.2.2 Access Control

Discretionary Access Control

6.2.2.1 [AC1] The product shall define and control discretionary access between subjects and objects. (See Section 6.1.2 for a definition of subjects and objects.)

6.2.2.2 [AC2] The product's definition and control of discretionary access between subjects and objects shall be implemented by the following two discretionary access control (DAC) mechanisms:

a) [AC2.1] access mode (owner/group/other) permissions

b) [AC2.2] access control lists (ACLs).

6.2.2.3 [AC3] ACLs shall only be applied to File System Objects, as follows:

a) for HFS File Systems using an HFS ACL

b) for JFS File Systems using a JFS ACL.

Access Mode Permissions

6.2.2.4 Each File System Object is associated with the following attributes:

a) an owning user identification (owner user ID)

b) a group identification (group ID)

c) a set of access permissions.

6.2.2.5 Each System V IPC and POSIX IPC object is associated with the following attributes:

a) an owning user identification (owner user ID)

b) a group identification (group ID)

c) (System V only) a creator user identification (creator user ID)

d) (System V only) a creator group identification (creator group ID)

e) a set of access permissions.

6.2.2.6 [AC8] 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:

a) [AC8.1] the *owner* of the object, identified by the owner user ID associated with the object

b) [AC8.2] any member of the *group* (identified by the group ID) associated with the object (except the owner)

c) [AC8.3] any *other* user (except the owner of the object or any member of the group associated with the object).

6.2.2.7 [AC9] 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:

a) [AC9.1] 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

b) [AC9.2] 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)

c) [AC9.3] any *other* user (except the owner or (System V only) creator of the object or any member of the group or (System V only) creator group associated with the object).

6.2.2.8 [AC10] The product shall allow selection of no access, or any combination of the access mode permissions specified in Table 6.1 for access to an object, independently for each class of user (owner, group, other).

Table 6.1 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	-	-	-	-

6.2.2.9 [AC12] Whenever 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 product, against the process effective user ID, effective group ID, and any group ID in the process's 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 AC13 and for System V IPC and POSIX IPC objects in AC14.)

6.2.2.10 [AC13] 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:

a) [AC13.1] 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

b) [AC13.2] 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

c) [AC13.3] the process's effective user ID does not match the object's owner user ID, the object group ID does not match the process's effective group ID or a group in the process's group access list, and the appropriate access mode permission is set for the object's *other* class of user

d) [AC13.4] the process has superuser status.

6.2.2.11 [AC14] '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:

a) [AC14.1] the process's effective user ID matches the object's owner user ID or (System V only) creator user ID and the appropriate access mode permission is set for the object's *owner* class of user

b) [AC14.2] 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 permission is set for the object's *group* class of user

c) [AC14.3] 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

d) [AC14.4] the process has superuser status.

6.2.2.12 [AC17] 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.

6.2.2.13 [AC18] When a process creates a new File System Object, the object's group ID is set:

a) [AC18.1] 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

b) [AC18.2] to the effective group ID of the process, if the set-group-ID attribute is not present in the parent directory's set of file protection attributes.

6.2.2.14 [AC19] 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*).

6.2.2.15 [AC22] A process shall be able to modify the access mode permissions associated with a File System Object, provided one or both of the following hold:

a) [AC22.1] the process has ownership rights to the object

b) [AC22.2] the process is privileged, having superuser status.

6.2.2.16 [AC23] 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:

a) [AC23.1] removed

b) [AC23.2] the process has ownership rights to the object and the process is a member of a privilege group allowing CHOWN

c) [AC23.3] the process is privileged, having superuser status.

6.2.2.17 [AC27] When a process creates a new System V IPC or 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.

6.2.2.18 [AC28] 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.

6.2.2.19 [AC30] A process shall be able to modify the access mode permissions associated with a System V IPC or POSIX IPC object, provided one or both of the following hold:

a) [AC30.1] the process has ownership rights, or (System V only) creator rights, or both ownership and (System V only) creator rights to the object

b) [AC30.2] the process is privileged, having superuser status.

6.2.2.20 [AC31] 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:

a) [AC31.1] the process has ownership rights, or (System V only) creator rights, or both ownership and (System V only) creator rights to the object

b) [AC31.2] the process is privileged, having superuser status.

HFS Access Control Lists

6.2.2.21 [AC36] Each HFS ACL entry shall specify for one user ID/group ID combination, a set of access permissions (as specified in Table 6.1) to the associated object, which may be zero or more of the following:

a) [AC36.1] read

b) [AC36.2] write

c) [AC36.3] execute/search.

6.2.2.22 [AC39] Whenever an unprivileged process makes a request to open a HFS Object, the ACL for that object shall be checked by the product's access check algorithm (AC40 and AC41) to determine whether the process can access the object in the requested mode.

6.2.2.23 [AC40] The product'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 userID/group ID combination, in the following order of precedence:

a) [AC40.1] specific user, specific group

b) [AC40.2] specific user, no specific group

c) [AC40.3] no specific user, specific group

d) [AC40.4] no specific user, no specific group.

6.2.2.24 [AC41] Where a process has more than one group ID, the product's access check algorithm shall set the access mode to the union of the permissions in all matching ACL entries of the same level of precedence.

6.2.2.25 [AC42] A process shall be able to modify the ACL associated with an object, provided one or both of the following hold:

a) [AC42.1] the process has ownership rights to the object

b) [AC42.2] the process has superuser status.

6.2.2.26 [AC43] When a process creates a new object, the product creates three base ACL entries to correspond with the object's access mode permissions (as determined by AC19) as follows:

a) [AC43.1] base ACL entry for the object's *owner* class of user

b) [AC43.2] base ACL entry for the object's *group* class of user

c) [AC43.3] base entry for the object's *other* class of user.

6.2.2.27 [AC44] The product 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.

JFS Access Control Lists

6.2.2.28 [AC52] Each JFS 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 Table 6.1) to the associated object, which may be zero or more of the following:

a) [AC52.1] read

b) [AC52.2] write

c) [AC52.3] execute/search.

6.2.2.29 [AC53] Whenever an unprivileged process makes a request to open a JFS Object, the ACL for that object shall be checked by the product's access check algorithm (AC54) to determine whether the process can access the object in the requested mode.

6.2.2.30 [AC54] The product'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 of precedence:

a) [AC54.1] permissions as specified in the *user* entry

b) [AC54.2] permissions as specified in the additional user entry, bitwise-AND'd with those in the *class* entry

c) [AC54.3] permissions as specified in the *group* entry

d) [AC54.4] permissions as specified in the additional group entry, bitwise-AND'd with those in the *class* entry

e) [AC54.5] permissions as specified in the *other* entry.

6.2.2.31 [AC55] A process shall be able to modify the ACL associated with an object, provided one or both of the following hold:

a) [AC55.1] the process has ownership rights to the object

b) [AC55.2] the process has superuser status.

6.2.2.32 [AC56] When a process creates a new object, the product creates four base ACL entries to correspond with the object's access mode permissions (as determined by AC19) as follows:

a) [AC56.1] base ACL entry for the object's *owner* class of user

b) [AC56.2] base ACL entry for the object's *group* class of user

c) [AC56.3] base ACL entry for the object's group *class*

d) [AC56.4] base entry for the object's *other* class of user.

6.2.2.33 [AC57] When a process creates a new object, the product creates ACL entries corresponding with any default ACL entries of the directory in which the object is created.

6.2.2.34 [AC58] The product 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.

Process Control

6.2.2.35 [AC45] Whenever a process is created, the product shall ensure that the following attributes are inherited from the parent process:

a) [AC45.1] the real user ID

b) [AC45.2] the real group ID

c) [AC45.3] the effective user ID

d) [AC45.4] the effective group ID

e) [AC45.5] the group access list

f) [AC45.6] the process's current working directory

g) [AC45.7] the audit ID.

6.2.2.36 [AC46] Whenever a session leader process is created, the product shall ensure that the process's attributes listed in AC45 are equal to those associated with the user authenticated during login, that is:

a) [AC46.1] the real and effective user IDs are set equal to the user's user ID

b) [AC46.2] the real and effective group IDs are set equal to the user's group ID

c) [AC46.3] the group access list is set equal to the set of supplementary group IDs

d) [AC46.4] the audit ID is set equal to the user's audit ID

e) [AC46.5] the current working directory is set equal to the user's home directory.

6.2.2.37 [AC47] Whenever an executable object is executed by a process, the product shall ensure that:

a) [AC47.1] 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

b) [AC47.2] 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.

6.2.2.38 [AC51] Only a superuser shall be able to change the real and effective user Ids of a process without re-authentication.

Policy Enforcement

6.2.2.39 [AC59] The product shall validate all attempted operations between subjects and objects, ensuring that all relevant DAC policy enforcement checks succeed before access is granted.

6.2.3 Audit

Audit Data Collection

6.2.3.1 [AUD1] The product shall be capable of auditing all security relevant events that occur as a result of actions performed by the product on behalf of a user (system calls), on a per event and per user basis.

6.2.3.2 [AUD2] The product shall allow only a superuser to turn the auditing capability on or off.

6.2.3.3 Note: Section 2.2 assumes that auditing is on when the product is operated in multi-user mode.

6.2.3.4 [AUD3] The product shall allow only a superuser 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.

6.2.3.5 [AUD4] The product shall protect the audit data so that it cannot be accessed by any user who is not authorised so to do.

Audit Events

6.2.3.6 [AUD5] The product shall group system calls having a similar behaviour into categories called 'event types'.

6.2.3.7 [AUD6] The product shall provide the event types listed in Table 6.2.

6.2.3.8 [AUD7] The product shall allow only a superuser to set or observe the auditing status of event types, on a per event type basis, to one of the following:

a) [AUD7.1] audit for success only

b) [AUD7.2] audit for failure only

c) [AUD7.3] audit for both success and failure

d) [AUD7.4] do not audit.

6.2.3.9 [AUD8] The product shall allow only a superuser to set or observe the auditing status of system calls, on a per system call basis, to one of the following:

a) [AUD8.1] audit for success only

b) [AUD8.2] audit for failure only

c) [AUD8.3] audit for both success and failure

d) [AUD8.4] do not audit.

6.2.3.10 [AUD9] This SF has been removed.

6.2.3.11 [AUD10] The product's initial default selection of audit events shall audit the success and failure of the following event types:

a) [AUD10.1] admin

b) [AUD10.2] logon

c) [AUD10.3] moddac.

Table 6.2 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), clock_settime(2), kload(2), mpctl(2), plock(2), reboot(2), sched_setparam(2), sched_setscheduler(2), serialize(2), setaudid(2), setaudproc(2), setdomainname(2), setevent(2), setprivgrp(2), setrlimit64(2), stime(2), swapon(2), toolbox(2), utssys(2)*
close	Log all closings of objects	*close(2), ksem_close(2), mq_close(2), munmap(2)*
create	Log all creations of objects	*creat(2), mkdir(2), mknod(2), msgget(2), pipe(2), semget(2), shmat(2), shmget(2), symlink(2)*
delete	Log all deletions of objects	*Ksem_unlink(2), mq_unlink(2), msgctl(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)*
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), link(2), lockf(2), lockf64(2), rename(2), setcontext(2), setgid(2), setgroups(2), setpgid(2), setpgrp(2), setregid(2), setresgid(2), setresuid(2), setsid(2), setuid(2), shmctl(2), shmdt(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), 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), vfsmount(2)*
uevent1, uevent2, uevent3	Log user defined events	See 'Streamlining Audit Log Data'

Streamlining Audit Log Data

6.2.3.12 [AUD12] The product shall provide the capability for superusers to program processes so that auditing of system calls may be suspended or resumed at appropriate points in the process (known as a self-auditing process) and an alternative or additional, single, audit event is produced.

6.2.3.13 [AUD13] The processes listed in Table 6.3 shall be self-auditing.

6.2.3.14 [AUD14] The product shall provide the following three event types, for use by superuser defined self-auditing processes, for which the auditing status may be set as specified in AUD7:

a) [AUD14.1] uevent1

b) [AUD14.2] uevent2

c) [AUD14.3] uevent3.

Table 6.3 Self-auditing Processes

Process	Description
chfn(1)	Change finger entry
chsh(1)	Change login shell
login(1)	The login utility
newgrp(1)	Change effective group
passwd(1)	Change password
audevent(1M)	Select events to be audited
audisp(1M)	Display the audit data
audsys(1M)	Start or halt the auditing system
audusr(1M)	Select users to be audited
fbackup(1M)	Selectively back up files
useradd(1M)	Add new user login account
userdel(1M)	Delete user login account
usermod(1M)	Modify user login account

Audit Records

6.2.3.15 [AUD15] The first time an audit event occurs in a process after an audit log file is selected for use, the product shall write a process ID identification record into the audit log file which shall contain the following information:

a) [AUD15.1] process ID

b) [AUD15.2] parent process ID

c) [AUD15.3] audit ID

d) [AUD15.4] real user ID

e) [AUD15.5] real group ID

f) [AUD15.6] effective user ID

g) [AUD15.7] effective group ID

h) [AUD15.8] device name.

6.2.3.16 [AUD16] For each event audited, the product shall record in the selected audit log file the following information:

a) [AUD16.1] the absolute date and time that the audited event completes

b) [AUD16.2] the event type

c) [AUD16.3] the process ID of the process that causes the event

d) [AUD16.4] the success or failure of the event

e) [AUD16.5] event specific information, if required, as specified in AUD17 and AUD18.

6.2.3.17 [AUD28] The date and time inserted into audit records shall be reliable.

6.2.3.18 [AUD17] 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.

6.2.3.19 [AUD18] 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.

Viewing Audit Logs

[AUD19] The product shall provide the capability for only the superuser to extract audit log data from a specified audit log file in accordance with one or more of the following selection criteria:

a) [AUD19.1] a given user name

b) [AUD19.2] a given terminal port

c) [AUD19.3] a given set of event types

d) [AUD19.4] a given set of system calls

e) [AUD19.5] successful events

f) [AUD19.6] failed events

g) [AUD19.7] the event date and time at which to start the extraction of audit log data

h) [AUD19.8] the event date and time at which to end the extraction of audit log data.

Maintaining Audit Log Files

6.2.3.20 [AUD20] The product shall collect audit records in:

a) [AUD20.1] a *primary log file*, which is used initially by the product

b) [AUD20.2] an optional (as selected by a superuser) *auxiliary log file*.

6.2.3.21 [AUD21] The product shall allow a superuser to specify the following audit parameters:

a) [AUD21.1] an Audit File Switch (AFS) size

b) [AUD21.2] the File Space Switch (FSS) size.

6.2.3.22 [AUD22] The product shall issue a warning on the console when the primary log file reaches a percentage, configurable by a superuser, of the AFS size or the FSS size.

6.2.3.23 [AUD23] When the AFS size or the FSS size is reached, the product shall attempt to switch to the auxiliary log file to collect audit records.

6.2.3.24 [AUD24] If no auxiliary log file exists, the product shall periodically issue a warning on the console.

6.2.3.25 [AUD25] 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.

6.2.3.26 [AUD26] When the file system(s) is(are) completely full, no audit records shall be collected, although a superuser shall be allowed to continue to carry out operations.

6.2.3.27 [AUD27] The maximum number of audit records lost during a system crash (except when the file system is full and a superuser continues to carry out operations) shall be one per process.

6.2.4 Object Reuse

[OR1] The product 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).

6.2.5 Protection Functions

6.2.5.1 [PF1] The product shall maintain control and data separation between TSF functions executing in kernel space and functions executing in user space.

6.2.5.2 [PF2] The product shall maintain control and data separation between processes executing in user space.

6.2.5.3 [PF3] The product shall allow a superuser to run a test utility to confirm that 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.

6.3 Required Security Mechanisms

6.3.1 The product implements a password-checking algorithm to enforce the constraints on user-generated passwords specified in security function I&A18. The password-checking algorithm ensures that passwords satisfy the Strength of Function claim of SoF-Medium.

6.3.2 The product implements a modified one way DES algorithm to satisfy the password encryption function specified in security function I&A23. The assessment of the Strength of Function of encryption algorithms is outside the scope of evaluation.

6.4 Assurance Measures

The assurance measures adopted to satisfy each of the EAL4 assurance requirements, as defined in [CC] Part 3, Section 6.2.4, Table 6.5, are summarised in Table 6.4.

Table 6.4 Satisfaction of EAL4 Assurance Requirements by Assurance Measures

EAL4 Assurance Components	Assurance Measures
ACM_AUT.1 Partial CM automation	This requirement is met by Configuration Control [CCNTL] and by Trusted Delivery [TD].
ACM_CAP.4 Generation support and acceptance procedures	This requirement is met by Configuration Control [CCNTL] and by up-to-date configuration lists.
ACM_SCP.2 Problem tracking CM coverage	This requirement is met by Configuration Control [CCNTL] and by up-to-date configuration lists.
ADO_DEL.2 Detection of modification	This requirement is met by Trusted Delivery [TD].
ADO_IGS.1 Installation, generation, and start-up procedures	This requirement is met by Read Before Installing or Updating to HP-UX 11i [README], Release Notes [REL1] & [REL2], Installation Guide [INSTALL], Software Distributor Administration Guide [SDAG], Managing Systems and Workgroups [MSW] and Using HP-UX [USING].
ADV_FSP.2 Fully defined external interfaces	This requirement is met by Functional Specification [FS], which references relevant [Man Pages].
ADV_HLD.2 Security enforcing high-level design	This requirement is met by High Level Design [HLD] and Architecture Summary Document [ASD].
ADV_IMP.1 Subset of the implementation of the TSF	This requirement is met by HP-UX 11.11 source code.
ADV_LLD.1 Descriptive low-level design	This requirement is met by Low Level Design Documents [LLD].
ADV_RCR.1 Informal correspondence demonstration	This requirement is met by the [FS], the [HLD] and the [LLD].
ADV_SPM.1 Informal TOE security policy model	This requirement is met by this document.
AGD_ADM.1 Administrator	This requirement is met by Managing Systems and Workgroups

Table 6.4 Satisfaction of EAL4 Assurance Requirements by Assurance Measures

EAL4 Assurance Components	Assurance Measures
guidance	[MSW], Evaluated Configuration Guide [ECG] and [Man Pages].
AGD_USR.1 User guidance	This requirement is met by Using HP-UX [USING], Evaluated Configuration Guide [ECG] and [Man Pages].
ALC_DVS.1 Identification of security measures	This requirement is met by Trusted Delivery [TD].
ALC_LCD.1 Developer defined life-cycle model	This requirement is met by Configuration Control [CCNTL].
ALC_TAT.1 Well-defined development tools	This requirement is met by Configuration Control [CCNTL].
ATE_COV.2 Analysis of coverage	This requirement is met by Test Plan [TPLAN].
ATE_DPT.1 Testing: high-level design	This requirement is met by Test Plan [TPLAN], together with the demonstration of correspondence between the [FS] and the [HLD] required by ADV_RCR.1.
ATE_FUN.1 Functional testing	This requirement is met by Test Plan [TPLAN], Test Procedures [TPROC], Security Test Journal [STJ], Security Test Report [STR] and Multi-Platform Rationale [MPR].
ATE_IND.2 Independent testing – sample	Representative platform(s) are provided to enable the evaluators to perform independent functional testing.
AVA_MSU.2 Validation of analysis	This requirement is met by Misuse Analysis [MSU].
AVA_SOF.1 Strength of TOE security function evaluation	This requirement is met by Strength of Function Analysis [SOF].
AVA_VLA.2 Independent vulnerability analysis	This requirement is met by Vulnerability Analysis [VA]. Representative platform(s) are provided to enable the evaluators to perform vulnerability testing.

7 PP Claims

7.1 PP Reference

The TOE conforms to [CAPP].

7.2 PP Tailoring

TOE security functional requirements derived from [CAPP] that have been tailored by performing the operations required by [CAPP] are defined in Section 5.1, with assignments and selections underlined.

7.3 PP Additions

There are no additions to the TOE security functional requirements or objectives derived from [CAPP].

8 Rationale

8.1 Security Objectives Rationale

The specification of security objectives for the TOE and the environment in Chapter 4 and the specification of the TOE security environment in Chapter 3 are fully conformant with [CAPP]. Therefore the security objectives rationale presented in [CAPP] Section 7.1 applies and is not repeated here.

8.2 Security Requirements Rationale

8.2.1 Security Functional Requirements Cover Security Objectives

The security functional requirements for the TOE comply with [CAPP], with no augmentation (see Section 5.1). The specification of security objectives for the TOE and the environment in Chapter 4 are fully conformant with [CAPP]. Therefore the rationale for 'complete coverage – objectives' in [CAPP] Section 7.2.2 applies and is not repeated here.

8.2.2 Internal Consistency of Requirements

The security functional requirements for the TOE comply with [CAPP], with the required operations of assignment and selection performed to make the requirements TOE specific. The assignment and selection operations were performed using consistent computer security and TOE specific terminology. Therefore the rationale for internal consistency of requirements presented in [CAPP] Section 7.2.1 applies and is not repeated here.

8.2.3 Satisfaction of Dependencies

The security functional requirements for the TOE comply with [CAPP], with no augmentation. Therefore the dependencies in [CAPP] Section 7.3, with the CC Identifier entry in the table corresponding to [CAPP] Section 5.4.6 corrected to read FMT_MTD.1.

8.2.4 Justification of Assurance Level

This security target is fully conformant with [CAPP] with no augmentation. [CAPP] was developed for a generalised environment with moderate risk to the assets and as such an assurance level of EAL3 was deemed to be appropriate. This security target claims an assurance level of EAL4, which is also appropriate.

8.2.5 Justification of Strength of Function Claim

This security target specifies in Section 5.2 a Strength of Function claim of SOF-Medium, which is consistent with the [CAPP] security functional requirement FIA_SOS.1, as justified in [CAPP] Section 7.5.

8.3 TOE Summary Specification Rationale

8.3.1 Satisfaction of TOE Security Functional Requirements

Table 8.1 demonstrates that the combination of specified TOE security functions work together to satisfy the TOE security functional requirements.

Table 8.1 Mapping of Security Functions to Security Functional Requirements

Security Functional Requirements		Security Functions
Element	**Component Name**	
FAU_GEN.1.1	Audit Data Generation	AUD1, AUD6
FAU_GEN.1.2	Audit Data Generation	AUD16, AUD17, AUD18
FAU_GEN.2.1	User Identity Association	AUD15, AUD16
FAU_SAR.1.1	Audit Review	AUD19
FAU_SAR.1.2	Audit Review	AUD19
FAU_SAR.2.1	Restricted Audit Review	AUD4, AUD19
FAU_SAR.3.1	Selectable Audit Review	AUD19
FAU_SEL.1.1	Selective Audit	AUD3, AUD5, AUD7, AUD8, AUD10, AUD12, AUD13, AUD14
FAU_STG.1.1	Guarantees of Audit Data Availability	AUD2, AUD4
FAU_STG.1.2	Guarantees of Audit Data Availability	AUD27
FAU_STG.3.1	Action in Case of Possible Audit Data Loss	AUD20, AUD21, AUD22
'FAU_STG.4.1'	Prevention of Audit Data Loss	AUD23, AUD24, AUD25, AUD26
FDP_ACC.1.1	Discretionary Access Control Policy	AC1, AC44, AC58
FDP_ACF.1.1	Discretionary Access Control Functions	AC2, AC3, AC8, AC9, AC10, AC36, AC52
FDP_ACF.1.2	Discretionary Access Control Functions	AC12, AC13, AC14, AC39, AC40, AC41, AC53, AC54
FDP_ACF.1.3	Discretionary Access Control Functions	AC13, AC14
FDP_ACF.1.4	Discretionary Access Control Functions	AC12, AC13, AC14, AC39, AC40, AC41, AC53, AC54
FDP_RIP.2[1].1	Object Residual Information Protection	OR1
'FDP_RIP.2[2].1' (Note 1)	Subject Residual Information Protection	OR1
FIA_ATD.1.1	User Attribute Definition	I&A1
FIA_SOS.1.1	Strength of Authentication Data	I&A8, I&A16, I&A18
FIA_UAU.1.1	Authentication	I&A6, I&A7, I&A8, I&A11
FIA_UAU.1.2	Authentication	I&A6, I&A7, I&A8, I&A11
FIA_UAU.7.1	Protected Authentication Feedback	I&A24

Table 8.1 Mapping of Security Functions to Security Functional Requirements

Security Functional Requirements		Security Functions
Element	**Component Name**	
FIA_UID.1.1	Identification	I&A6, I&A7, I&A11, I&A13
FIA_UID.1.2	Identification	I&A6, I&A7, I&A11, I&A13
'FIA_USB.1.1' (Note 2)	User-Subject Binding	AC45
'FIA_USB.1.2' (Note 2)	User-Subject Binding	I&A13, I&A14, AC46
'FIA_USB.1.3' (Note 2)	User-Subject Binding	AC47, AC51
FMT_MSA.1.1	Management of Object Security Attributes	AC22, AC23, AC30, AC31, AC42, AC55
FMT_MSA.3.1	Static Attribute Initialisation	AC17, AC18, AC19, AC27, AC28, AC43, AC56, AC57
FMT_MSA.3.2	Static Attribute Initialisation	AC22, AC23, AC30, AC31, AC42, AC55
FMT_MTD.1^1.1	Management of the Audit Trail	AUD2, AUD3, AUD4, AUD20
FMT_MTD.1^2.1	Management of Audited Events	AUD3, AUD4, AUD7, AUD8, AUD12, AUD19
FMT_MTD.1^3.1	Management of User Attributes	I&A5
FMT_MTD.1^4.1^1	Management of Authentication Data	I&A5
FMT_MTD.1^4.1^2	Management of Authentication Data	I&A5, I&A23, I&A24
FMT_REV.1^1.1	Revocation of User Attributes	I&A5
FMT_REV.1^1.2	Revocation of User Attributes	I&A5
FMT_REV.1^2.1	Revocation of Object Attributes	AC22, AC42, AC55
FMT_REV.1^2.2	Revocation of Object Attributes	AC12, AC13, AC14, AC39, AC40, AC41, AC53, AC54
FMT_SMR.1.1	Security Management Roles	I&A5, I&A16, AC13, AC14, AC22, AC23, AC30, AC31, AC42, AC55, AUD2, AUD3, AUD4, AUD7, AUD8, AUD12, AUD19, AUD20, AUD21, AUD22, AUD26
FMT_SMR.1.2	Security Management Roles	I&A5, AC23
FPT_AMT.1.1	Abstract Machine Testing	PF3
FPT_RVM.1.1	Reference Mediation	AC59
FPT_SEP.1.1	Domain Separation	PF1
FPT_SEP.1.2	Domain Separation	PF2

Table 8.1 Mapping of Security Functions to Security Functional Requirements

Security Functional Requirements		Security Functions
Element	**Component Name**	
FPT_STM.1.1	Reliable Time Stamps	AUD28

8.3.2 Justification of Compliance with Assurance Requirements

The compliance of assurance measures with assurance requirements is demonstrated in Section 6.4.

8.4 PP Claims Rationale

This security target is fully conformant with [CAPP] with no augmentation and with the required operations of assignment and selection performed to make the requirements TOE specific.