

---

## **Security Target**


### **Tru64 UNIX (Version 5.1A)**



Prepared by: Ralph Worswick

Project Manager: David Surman-Roberts  
Compaq Computer Limited  
Worton Grange  
Imperial Way  
Reading  
Berkshire  
RG2 0TE

Project Document Id:  
Date Prepared: January 2004

Tru64 UNIX (Version 5.1A)	Security Target	
---------------------------	-----------------	---

**Document Information**

<b>Project Name:</b>	Tru64 UNIX (Version 5.1A) EAL1 Evaluation		
<b>Project Manager:</b>	David Surman-Roberts	<b>Document Version No:</b>	07
<b>FocusPM Phase:</b>		<b>Document Version Date:</b>	28/01/2004
<b>Quality Review Method:</b>	Peer Review		
<b>Prepared By:</b>	Ralph Worswick	<b>Preparation Date:</b>	28/01/2004
<b>Reviewed By:</b>	EAL1 Team	<b>Review Date</b>	

**Distribution List**


From	Date	Phone/Fax
Ralph Worswick	28/01/2004	01276 687314

To	Action*	Date	Phone/Fax
LogicaCMG	Inform	28/01/2004	
David Surman-Roberts	Review	28/01/2004	
Dennis McMann	Review	28/01/2004	
Jeanne Donnelly	Review	28/01/2004	
Ann Majeske	Review	28/01/2004	

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

**Version History**

Ver. No.	Ver. Date	Revised by	Description	Filename
01	02/03	RW	First Draft	Tru64 ST Draft 01
02	04/03	RW	Second Draft	Tru64 ST Draft 02
03	05/03	RW	Third Draft	Tru64 ST Draft 03
04	05/03	RW	Fourth Draft	Tru64 ST Draft 04
05	05/03	RW	Fifth Draft	Tru64 ST Draft 05
06	01/04	RW	First Issue	Tru64 ST Issue 1.0

Tru64 UNIX (Version 5.1A)	Security Target	 i n v e n t
---------------------------	-----------------	--

## **Proprietary Notice**

This document is the property of Hewlett-Packard Ltd (HP). All information herein is confidential to HP and must not be copied or disclosed to any third party without the prior written consent of HP.

Copyright © Hewlett-Packard Ltd.

# Contents

## References

<b>1</b>	<b>INTRODUCTION.....</b>	<b>1</b>
1.1	ST IDENTIFICATION .....	1
1.2	ST OVERVIEW .....	1
1.3	CC CONFORMANCE .....	1
1.4	DOCUMENT STRUCTURE .....	2
1.5	CONVENTIONS.....	2
1.6	TERMINOLOGY.....	4
<b>2</b>	<b>TOE DESCRIPTION .....</b>	<b>5</b>
2.1	PRODUCT TYPE .....	5
2.2	EVALUATED CONFIGURATION.....	5
2.3	SUMMARY OF SECURITY FEATURES .....	5
<b>3</b>	<b>TOE SECURITY ENVIRONMENT.....</b>	<b>7</b>
3.1	ASSUMPTIONS.....	7
3.2	THREATS.....	7
3.3	ORGANISATIONAL SECURITY POLICIES.....	7
<b>4</b>	<b>SECURITY OBJECTIVES .....</b>	<b>9</b>
4.1	SECURITY OBJECTIVES FOR THE TOE.....	9
4.2	SECURITY OBJECTIVES FOR THE ENVIRONMENT .....	9
<b>5</b>	<b>SECURITY REQUIREMENTS .....</b>	<b>10</b>
5.1	TOE SECURITY FUNCTIONAL REQUIREMENTS.....	10
5.2	TOE SECURITY ASSURANCE REQUIREMENTS.....	15
5.3	SECURITY REQUIREMENTS FOR THE IT ENVIRONMENT .....	15
<b>6</b>	<b>TOE SUMMARY SPECIFICATION .....</b>	<b>16</b>
6.1	CONCEPTS AND TERMINOLOGY .....	16
6.2	TOE SECURITY FUNCTIONS .....	18
6.3	REQUIRED SECURITY MECHANISMS.....	28
6.4	ASSURANCE MEASURES.....	28
<b>7</b>	<b>RATIONALE.....</b>	<b>30</b>
7.1	SECURITY OBJECTIVES RATIONALE.....	30
7.2	SECURITY REQUIREMENTS RATIONALE .....	30
7.3	TOE SUMMARY SPECIFICATION RATIONALE .....	31

## References

- [AdvFS] Tru64 AdvFS Administration  
([http://h30097.www3.hp.com/docs/base\\_doc/DOCUMENTATION/V51A\\_HTML/ARH96CTE/TITLE.HTM](http://h30097.www3.hp.com/docs/base_doc/DOCUMENTATION/V51A_HTML/ARH96CTE/TITLE.HTM))
- [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, CCIMB-99-031  
Part 2 Security functional requirements, CCIMB-99-032  
Part 3 Security assurance requirements, CCIMB-99-033
- [CMND\_SHELL] Tru64 UNIX Command and Shell User's Guide  
([http://h30097.www3.hp.com/docs/base\\_doc/DOCUMENTATION/V51A\\_HTML/ARH91BTE/TITLE.HTM](http://h30097.www3.hp.com/docs/base_doc/DOCUMENTATION/V51A_HTML/ARH91BTE/TITLE.HTM))
- [EVAL\_CONF] Tru64 Evaluated Configuration  
([http://h30097.www3.hp.com/docs/best\\_practices/sec\\_bps.html](http://h30097.www3.hp.com/docs/best_practices/sec_bps.html)) {Updated for Version 5.1A}
- [FS] Tru64 UNIX V5.1a Common Criteria Functional Specification, Hewlett-Packard, {Version & Date TBD}
- [INSTALL] Tru64 UNIX Installation Guide  
([http://h30097.www3.hp.com/docs/base\\_doc/DOCUMENTATION/V51A\\_HTML/ARH8SDTE/TITLE.HTM](http://h30097.www3.hp.com/docs/base_doc/DOCUMENTATION/V51A_HTML/ARH8SDTE/TITLE.HTM))
- [ITSEC ST] Tru64 UNIX (Version 4.0G) Security Target, Compaq, Issue 2.3, 19 June 2001
- [REF\_PAGES] Reference (Volumes 1 to 9)  
([http://h30097.www3.hp.com/docs/base\\_doc/DOCUMENTATION/V51A\\_HTML/REF\\_LIB.HTM](http://h30097.www3.hp.com/docs/base_doc/DOCUMENTATION/V51A_HTML/REF_LIB.HTM))
- [SECURITY] Tru64 UNIX Security  
([http://h30097.www3.hp.com/docs/base\\_doc/DOCUMENTATION/V51A\\_HTML/ARH95DTE/TITLE.HTM](http://h30097.www3.hp.com/docs/base_doc/DOCUMENTATION/V51A_HTML/ARH95DTE/TITLE.HTM))
- [SYS\_ADMIN] Tru64 UNIX System Administration  
([http://h30097.www3.hp.com/docs/base\\_doc/DOCUMENTATION/V51A\\_HTML/ARH9FDTE/TITLE.HTM](http://h30097.www3.hp.com/docs/base_doc/DOCUMENTATION/V51A_HTML/ARH9FDTE/TITLE.HTM))
- [TCSEC] Department of Defense Trusted Computer System Evaluation Criteria (TCSEC), National Computer Security Center, DOD 5200.28-STD, December 1985
- [X\_WINDOW] Tru64 UNIX X Window System Environment  
([http://h30097.www3.hp.com/docs/base\\_doc/DOCUMENTATION/V51A\\_HTML/ARH9JBTE/TITLE.HTM](http://h30097.www3.hp.com/docs/base_doc/DOCUMENTATION/V51A_HTML/ARH9JBTE/TITLE.HTM))
- [UPDATE] Technical Update for Tru64 UNIX Version 5.1A at  
<http://h30097.www3.hp.com/docs/updates/V5.1a/TITLE.HTM>



# 1 Introduction

## 1.1 ST Identification

1.1.1 Title: Security Target for Tru64 UNIX (Version 5.1A).

1.1.2 Keywords: Tru64, UNIX, POSIX, general purpose operating system.

1.1.3 This document is the Security Target for Tru64 UNIX (Version 5.1A), a general purpose UNIX operating system product offered by Hewlett-Packard. 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 Tru64 UNIX (Version 5.1A), a general purpose UNIX operating system product offered by Hewlett-Packard, referred to as the 'product' and, in its evaluated configuration, the 'target of evaluation (TOE)', as submitted for evaluation to the [CC] evaluation assurance level EAL1.

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 – boot authentication and time-based logon restrictions. The [TCSEC] Class C2 requirements are described for [CC] in the Controlled Access Protection Profile [CAPP]. Security functional requirements in this Security Target are derived from [CAPP].

1.2.3 The TOE executes on a single HP Alphaserver, with direct access provided via the console and a local network.

## 1.3 CC Conformance

1.3.1 The TOE is [CC] Part 2 extended, Part 3 conformant with a claimed assurance level of EAL1.

1.3.2 No conformance with any Protection Profile is claimed. However, the following aspects of the Security Target are derived from, and in many cases identical with, those specified in [CAPP]:

- a) Assumptions
- b) Organisational security policies
- c) Security objectives for the TOE
- d) Security objectives for the environment
- e) TOE security functional requirements
- f) Security objectives rationale
- g) Security requirements rationale.

1.3.3 The reason that [CAPP] conformance is not claimed is that:

- a) [CAPP] requires an Evaluation Assurance Level of at least EAL3, whereas this Security Target claims an Evaluation Assurance Level of EAL1
- b) Assumption A.NETWORK is modified to allow authenticated user connection using ftp and telnet services via unspecified clients that are assumed not to implement the TOE security policy.

## 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 rationale.

## 1.5 Conventions

- 1.5.1 Security functional requirements specified in Section 51, 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, where the label is a group of characters representing an aspect of security followed by a sequence number (e.g. I&A1, representing the first security function for the Identification and Authentication aspect).
- 1.5.3 Security functions are derived mainly from the security enforcing functions (SEFs) specified in the Security Target [ITSEC ST] for the E2 evaluated version of Tru64 UNIX Version 4.0G and are identical to the SEFs in many cases. Modifications to [ITSEC ST] SEFs are presented in underlined text. Some of the [ITSEC ST] SEFs have been excluded from the scope of this evaluation. Table 1.1 shows the correspondence between the security functions specified for the TOE in this Security Target and the SEFs specified for Tru64 UNIX Version 4.0G in [ITSEC ST].
- 1.5.4 Document references are identified as abbreviations in square brackets.

<b>Table 1.1 Correspondence between TOE SFs and Tru64 UNIX Version 4.0G SEFs/Mechanisms</b>	
<b>TOE SF or Comment</b>	<b>Tru64 UNIX Version 4.0G SEFs/Mechanisms and/or Comment</b>
I&A1	SEF1.1a
I&A2	SEF1.1b
I&A3	New security function
I&A4	New security function
I&A5	SEF1.2
I&A6	SEF1.3 (modified to clarify user's unique identity is the audit id)
I&A7	SEF1.4 (modified to extend to TOE maintenance of user group memberships)
I&A8	SEF1.5



I&A9	New security function
I&A10	New security function
I&A11	New security function
I&A12	New security function
I&A13	SEF1.7
I&A14	SEF1.8
I&A15	New security function
I&A16	New security function
Excluded	SEF1.9a, SEF 1.9b, SEF1.10, SEF1.11a, SEF1.11b, SEF1.12a, SEF1.12b
I&A17	SEF1.13
Excluded	SEF1.16, SEF1.17
AC1	SEF4.1
AC2	SEF4.2a
AC3	SEF4.2b
AC4	SEF4.2c
AC5	SEF4.3a
AC6	SEF4.3b (modified to cover groups as well as users)
AC7	SEF4.4 (modified to reference the applicable access check rules)
AC8	SEF4.6
AC9	SEF4.7a
AC10	SEF4.7b
AC11	SEF4.8 (modified to reference the applicable access check rules)
AC12	SEF4.10
AUD1	SEF5.1
AUD2	SEF5.2
AUD3	SEF5.3
AUD4	SEF5.4 (modified to explicitly include the event type and audit id in the available audit data)
AUD5	New security function
AUD6	SEF5.5
AUD7	SEF5.6
AUD8	SEF5.7
AUD9	SEF5.8 (modified to write a message and shutdown when 90% disk usage threshold reached)

OR1	SEF6.1
OR2	SEF6.2
OR3	SEF6.3
OR4	SEF6.4
PF1	M_4
PF2	M_5
PF3	New security function
Excluded	M_1, M_2, M_3, M_6, M_7, M_8, M_9

## 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 super-user, having an effective user identity (euid) of zero.

## 2 TOE Description

### 2.1 Product Type

- 2.1.1 The product is a high-performance, multi-user UNIX operating system to be used for general purpose computing services. The TOE executes on a single HP Alphaserver.
- 2.1.2 The target environment for the TOE is as a server providing applications support and secure file storage. Within the scope of this evaluation the only direct access to the TOE is via the console and a local network.
- 2.1.3 The TOE offers ftp and telnet services, constrained by the TOE's user authentication security functions, to unspecified clients connected via the network interfaces offered by the TOE. Note that the unspecified clients and the network link itself are considered to be outside the scope of evaluation and are not assumed to implement the TOE security policy.
- 2.1.4 The TOE is an 'evaluated configuration' of the product, as defined in Section 2.2.

### 2.2 Evaluated Configuration

The evaluated configuration of the product is defined as follows:

- a) Tru64 UNIX Version 5.1A with patch kit `common_criteria_cert_t64v51a` executing on a single, HP AlphaServer hardware platform selected from:  
AS300, 400, 800, 1000, 1000A, 1200, 2100, 2100A, 4000, 4100, 8200, 8400, DS10, DS10L, DS20, DS20E, ES40, ES45, GS60E, GS80, GS140, GS160, and GS320. Note that the GS80, GS160 and GS320 are limited to single partition use only.
- b) The software must be configured as described in [EVAL\_CONF], which details the installation of the operating system, post-installation setup of authentication, auditing, and other security-related items, installation of the patch kit, and the establishment of customized parameters for system and network daemons.
- c) Once configured, the system should be operated and maintained in accordance with the instructions, recommendations and guidance for secure operation given in Appendix E, C2 Level Security Configuration, of [SECURITY].

### 2.3 Summary of Security Features

#### 2.3.1 Introduction

The main security features of the TOE 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 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.
- 2.3.2.2 The TOE supports an ordinary *user* role and a *super-user* (administrative) role.
- 2.3.2.3 A super-user has 'root privilege' and is not constrained by the TOE's security policies.

2.3.2.4 An ordinary user does not have ‘root privilege’ and is constrained by the TOE’s security policies.

2.3.2.5 The authentication features are supported by user account locking after a configurable number of failed identification and authentication attempts.

### **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 TOE implements a DAC policy, which provides both the traditional UNIX ‘owner’, ‘group’, ‘world’ access mode permissions and a more granular access control list (ACL) mechanism, controlled by the object’s owner.

2.3.3.3 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 TOE is capable of collecting audit records for all security relevant events that occur. A super-user 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 super-user selectively for any period on the basis of criteria such as user identity, event type and object identity.

2.3.4.3 Facilities are provided to enable the super-user 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 identical with those specified in [CAPP], except for A.CONNECT, which is made TOE specific, and A.NETWORK, which replaces A.PEER. The assumptions of [CAPP] retained in this Security Target apply to the TOE as a stand alone server; the assumption A.NETWORK extends the TOE scope to permit connection by unspecified networked clients.

#### **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.NETWORK**

The TOE offers ftp and telnet services, constrained by the TOE's user authentication security functions, to unspecified clients connected via the network interfaces offered by the TOE. The unspecified clients and the network link are not assumed to implement the TOE security policy.

##### **A.CONNECT**

All connections to peripheral devices reside within the controlled access facilities. The TOE only addresses 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.

### **3.3 Organisational Security Policies**

The organisational security policies are identical with those specified in [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 identical with those specified in [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 identical with those specified in [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 <sup>&</sup> 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 <sup>&amp;</sup>	5.1.5.1
FAU_SEL.1	Selective Audit	FAU_SEL.1.1 <sup>&amp;</sup>	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 <sup>&amp;</sup>	5.1.8.1
‘FAU_STG.4’	Prevention of Audit Data Loss	‘FAU_STG.4.1’ <sup>&amp;</sup>	5.1.9.1
FDP_ACC.1	Discretionary Access Control Policy	FDP_ACC.1.1 <sup>&amp;</sup>	5.2.1.1
FDP_ACF.1	Discretionary Access Control Functions	FDP_ACF.1.1 <sup>&amp;</sup> FDP_ACF.1.2 <sup>&amp;</sup> FDP_ACF.1.3 <sup>&amp;</sup> FDP_ACF.1.4 <sup>&amp;</sup>	5.2.2.1 5.2.2.2 5.2.2.3 5.2.2.4
FDP_RIP.2 <sup>1</sup>	Object Residual Information Protection	FDP_RIP.2 <sup>1</sup> .1	5.2.3.1
‘FDP_RIP.2 <sup>2</sup> ’ (Note 1)	Subject Residual Information Protection	‘FDP_RIP.2 <sup>2</sup> .1’ (Note 1)	5.2.4.1
FIA_ATD.1	User Attribute Definition	FIA_ATD.1.1 <sup>&amp;</sup>	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 <sup>&amp;</sup> 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 <sup>&amp;</sup> FIA_UID.1.2	5.3.5.1 5.3.5.2
'FIA_USB.1' (Note 2)	User-Subject Binding	'FIA_USB.1.1' <sup>&amp;</sup> 'FIA_USB.1.2' <sup>&amp;</sup> 'FIA_USB.1.3' <sup>&amp;</sup> (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 <sup>&amp;</sup>	5.4.1.1
FMT_MSA.3	Static Attribute Initialisation	FMT_MSA.3.1 FMT_MSA.3.2 <sup>&amp;</sup>	5.4.2.1 5.4.2.2
FMT_MTD.1 <sup>1</sup>	Management of the Audit Trail	FMT_MTD.1 <sup>1</sup> .1	5.4.3.1
FMT_MTD.1 <sup>2</sup>	Management of Audited Events	FMT_MTD.1 <sup>2</sup> .1	5.4.4.1
FMT_MTD.1 <sup>3</sup>	Management of User Attributes	FMT_MTD.1 <sup>3</sup> .1	5.4.5.1
FMT_MTD.1 <sup>4</sup>	Management of Authentication Data	FMT_MTD.1 <sup>4</sup> .1 <sup>1</sup> FMT_MTD.1 <sup>4</sup> .1 <sup>2</sup>	5.4.6.1 5.4.6.2
FMT_REV.1 <sup>1</sup>	Revocation of User Attributes	FMT_REV.1 <sup>1</sup> .1 FMT_REV.1 <sup>1</sup> .2 <sup>&amp;</sup>	5.4.7.1 5.4.7.2
FMT_REV.1 <sup>2</sup>	Revocation of Object Attributes	FMT_REV.1 <sup>2</sup> .1 FMT_REV.1 <sup>2</sup> .2 <sup>&amp;</sup>	5.4.8 5.4.8.1
FMT_SMR.1	Security Management Roles	FMT_SMR.1.1 <sup>&amp;</sup> FMT_SMR.1.2	5.4.9.1 5.4.9.2
FPT_AMT.1	Abstract Machine Testing	FPT_AMT.1.1 <sup>&amp;</sup>	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) Audit event type;
- c) Identity of objects accessed;
- d) Time interval.

- 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) Audit event type.
- FAU\_STG.3.1 The TSF shall generate an alarm to the authorized administrator if the audit trail exceeds ninety percent 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 objects and non-file 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 file objects, the Access Control List (ACL);
    - ii) For non-file objects, the owner/group/world access 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 file objects, the ACL for the object is checked as follows:
    - i) If the process is the owner of the object, the permissions in the owning user:: entry are granted. Any other ACL entries are not checked;
    - ii) If the euid of the process matches a uid listed in a user: entry or resolves to a username listed in a user: entry, the permissions in the entry are granted. Any remaining ACL entries are not checked;
    - iii) If the egid of the process matches the gid of the file, or if one of the supplementary groups of the process matches the gid of the file, the process is granted the union of the permissions of the group: entry and any matching group: entries as described in the next list item;
    - iv) If the egid of the process matches the gid of any group: entries, or resolves to a groupname listed in any group: entries or if the gid or groupname of any of the supplementary groups of the process match any group: entries of the ACL, the process is granted the union of the protections of all matching group entries;
    - v) The permissions in the other: entry are granted.
  - b) For non-file objects, the owner/group/world access permissions are checked as follows:
    - i) If the euid of the process matches the object's uid or cuid, the object's owner permissions are granted – group and world permissions are ignored;
    - ii) If the egid of the process, or any gid in the process' group access list, matches the object's gid or cgid, the object's group permissions are granted – world permissions are ignored;
    - iii) The object's world permissions are granted.

- 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 super-user (with an euid 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 the absence of permissions in specific ACL entries of file objects checked as specified in FDP\_ACF.1.2a.
- 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) Login Name;
  - f) Audit Mask
  - g) Minimum Password Length.
- 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 per-user audit mask, which specifies the auditing of specific events on a user-by-user basis.
- '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 login User Identifier;
  - 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 subject's audit mask is set to the user's Audit Mask.
- '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 euid 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 ACL entries of file objects and in the owner/group/world access permissions of non-file objects;
  - b) An authorised administrator acting as superuser (with an euid 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<sup>1</sup>.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<sup>2</sup>.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 TOE Security Assurance Requirements**

The TOE security assurance requirements are those of evaluation assurance level EAL1 with no augmentation or extension.

## **5.3 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 identity (pid)
  - b) the parent process identity (ppid)
  - c) the process group identity (pgid)
  - d) the process's real and effective user identities (ruid & euid)
  - e) the process's real and effective group identities (rgid & egid)
  - f) a group access list – a list of groups to which the subject has access
  - g) an audit identity (audit id).
- 6.1.1.4 A user gains initial access to the product via login at a console, 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 ppid is set equal to its pid. All other processes in the same session share the same pgid. A process's ppid is the pid 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 ruid and euid are set equal to the user's user identity (uid)
  - b) the rgid and egid are set equal to the user's group identity (gid)
  - c) the group access list is set equal to the set of supplementary gids
  - d) the audit id is set equal to the user's login user identity (uid).
- 6.1.1.7 All security relevant attributes of a process (except the pid, ppid and pgids) 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 The association of a user with each process is the means by which the TOE holds users accountable. The pid uniquely identifies a process to the TOE. The audit id irrevocably associates the process and all of its descendants with a particular user and once initialised it can never be changed. The ruid and euid are normally set to the same values, but may be different under certain circumstances, for example in **setuid** programs. Similarly, the rgid and egid are normally set to the same values, but may be different under certain circumstances, for example in **setgid** programs.
- 6.1.1.10 A process running a **setuid** program has an euid of the owner of the program. For example, the **passwd(1)** command is **setuid** and the owner of the file /usr/bin/passwd is 'root'. When a user runs **passwd(1)** the process acting on behalf of the user has an euid of 'root', which enables it to change the password in the Protected Password Authentication database. Similarly **setgid** programs have an egid of the program's owning group. This 'privileged' mode is not retained after the **setuid** or **setgid** program exits. **setuid** and **setgid** programs enable users to perform privileged operations in a controlled manner. Note that the user's actions are still accountable when running a **setuid** or **setgid** program because the audit id (which never changes) is audited.
- 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) Super-user status, that is, a process executing with an euid of zero, equivalent to the 'root' user.
- 6.1.1.13 A process with superuser status is not constrained by the product's security policies.
- 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 basic objects implemented by the TOE are:
- a) file object:
    - i) directory
    - ii) regular file
    - iii) named pipe (FIFO file)
    - iv) symbolic link
    - v) (device) special file (character and block)
  - b) non-file objects:
    - i) process
    - ii) inter-process communication object:
      - shared memory
      - message queues
      - semaphore set

- unnamed pipe
  - UNIX domain socket (datagrams and streams)
  - INET domain socket (not included in evaluated configuration)
- iii) pseudo-terminal.

6.1.2.4 In addition, there are a number of objects associated with the DECwindows mandatory subsets of Tru64 UNIX. There are no means by which a user could defeat the security objectives of the TOE by attacking the contents of these objects because:

- a) an evaluated configuration of the TOE will not offer DECwindows services to remote users
- b) logon at the console will be restricted to individuals granted specific permission. These individuals are deemed to have the clearance to view all data on the system, and are trusted to uphold the confidentiality of data stored by the TOE (see [EVAL\_CONF]).

6.1.2.5 Access controls to protect DECwindows objects are deemed outside the scope of this evaluation.

6.1.2.6 The TOE implements a discretionary access control mechanism which classifies any process requesting an action to a non-file object into one of three categories (owner, group, world) based on comparison of the process's effective uid and effective gid with the identification of the object. Permission to exercise any combination of the three access modes is specified independently for each category.

6.1.2.7 Each file object has an access control list (ACL). POSIX 1003.6 Draft 13 ACLs are implemented in the system and are conceptually integrated with the mode bits, in that the mode bits are expressed as an ACL.

### **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 authentication, whereby even root (or privileged) users accessing the product are authenticated.

6.1.3.3 All subsequent accesses 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 Attributes relating to user authentication are stored in the Protected Password Authentication database. The following attributes relevant to the security functions, derived from **prpasswd(4)**, are stored for each user:
- a) login name
  - b) uid
  - c) encrypted password
  - d) audit mask
  - e) minimum password length.
- 6.2.1.2 Attributes relating to user group membership are stored in the `/etc/passwd` and `/etc/group` files.
- 6.2.1.3 The following attributes relevant to the security functions, derived from **passwd(4)**, are stored for each user:
- a) login name
  - b) login group id (gid).
- 6.2.1.4 The following attributes relevant to the security functions, derived from **group(4)**, are stored for each group:
- a) group name
  - b) gid
  - c) list of users allowed in the group.

#### **Password Authentication**

- 6.2.1.5 The TOE requires that a user identify himself to the TOE with a user name and password (i.e. log in) before allowing any other actions. When a user has been successfully authenticated, each process created is stamped with the audit id equal to the user's identifier (uid). The audit id cannot be changed once it is set.
- 6.2.1.6 I&A1: The TOE shall require users to uniquely identify and authenticate themselves to it before performing any other actions.
- 6.2.1.7 I&A2: This identification shall include the use of a user name (mapped internally to a user ID) and a password. The password shall not be echoed back to the screen.
- 6.2.1.8 I&A3: The user account shall be locked if the number of consecutive unsuccessful attempts to login to the user account exceeds the maximum allowed<sup>1</sup>, thereby preventing further successful logins to the account until it is unlocked by the super-user.
- 6.2.1.9 I&A4: The TOE's response to an authentication attempt with an invalid user name shall be the same as the response with a valid user name and invalid password.
- 6.2.1.10 I&A5: The TOE shall be able to enforce individual accountability by providing the capability to identify uniquely each individual user.
- 6.2.1.11 I&A6: The TOE shall provide the capability to associate a user's unique identity (audit id) with all auditable actions performed by that user.

---

<sup>1</sup> **Application Note:** [EVAL\_CONF] requires the TOE to be configured such the maximum permitted unsuccessful attempts to login to a user account before the account is locked is set to less than or equal to 5 (the default).

### User Security Attributes

- 6.2.1.12 I&A7: The TOE shall maintain authentication data identified in Paragraph 6.2.1.1 and user group membership data identified in Paragraphs 6.2.1.3 and 6.2.1.4<sup>2</sup>.
- 6.2.1.13 I&A8: The user authentication data shall be used by the TOE to authenticate the user's identity and to determine the security-relevant attributes of processes that may be created to act on behalf of the individual user.
- 6.2.1.14 I&A9: Whenever a process is created, the TOE shall ensure that the following attributes are inherited from the parent process:
- a) the real and effective user identities (ruid & euid)
  - b) the real and effective group identities (rgid & egid)
  - c) the group access list
  - d) the audit identity (audit id)
  - e) the per-user audit mask.
- 6.2.1.15 I&A10: Whenever a session leader process is created, the TOE shall ensure that the process's attributes listed in I&A9 are equal to those associated with the user authenticated during login, that is:
- a) the ruid and euid are set equal to the user's user identity (uid)
  - b) the rgid and egid are set equal to the user's group identity (gid)
  - c) the group access list is set equal to the set of supplementary groups (gids)
  - d) the audit id is set equal to the user's login user identity (uid)
  - e) the per-user audit mask is set equal to the user's audit mask.
- 6.2.1.16 I&A11: Only a super user shall be able to change the ruid and euid of a process without re-authentication.
- 6.2.1.17 I&A12: Whenever an executable object is executed by a process, the TOE shall ensure that:
- a) the process euid is set to the executable object's owner, if the **setuid** access mode is associated with the executable object
  - b) the process egid is set to the executable object's group, if the **setgid** access mode is associated with the executable object.

### Protected Encrypted Passwords

- 6.2.1.18 I&A13: The authentication data shall not contain a clear text version of each user's password, but rather a one-way encrypted value based on the user's password. When a user enters his password, it shall be used to construct an identically encrypted value which is compared against the encrypted value in the authentication data.
- 6.2.1.19 I&A14: The authentication data shall be protected so that it can only be directly modified by the super-user. Users shall be able to modify their password via trusted programs. Changes made shall take immediate effect.

---

<sup>2</sup> Application Note: [EVAL\_CONF] requires there to be a unique mapping between each user name and corresponding uid and between each group name and corresponding gid.

6.2.1.20 I&A15: Only a super user shall be permitted to set initial passwords.

#### **Password Generation**

6.2.1.21 I&A16: The TOE shall allow users to create user-generated passwords<sup>3</sup>.

6.2.1.22 I&A17: It shall be possible to configure the TOE to ensure that all passwords are greater than a minimum length determined by the super-user<sup>4</sup>.

#### **6.2.2 Access Control**

6.2.2.1 There is at least one uid and one gid associated with every object that is subject to the administration of access rights. The object's uid determines who owns it, in that any process whose effective uid equals the object's uid is considered to have ownership rights to the object. Both uids and gids are also considered in making discretionary access control decisions with respect to the object. When an object is created, its uid is taken from the effective uid of the creating process. For file-system objects, the gid is taken from the parent directory.

6.2.2.2 Some inter-process communication (IPC) objects have two sets of uids and gids which indicate the identity of the process which created them in addition to the current owner and owning group. Ownership rights to these IPC objects are granted to a process whose effective uid equals either of the object's uids, and both sets of uids and gids are considered in making discretionary access control decisions.

6.2.2.3 Objects are associated with permissions which are used by the discretionary access control mechanisms to grant or deny access by subjects to objects based on the uid and gid each held by subject and object.

#### **Default Permissions**

6.2.2.4 AC1: The DAC mechanisms shall ensure that when an object is created, it is assigned a set of default permissions which protect, or can be configured to protect, the object from unauthorised access.

#### **Discretionary (Need-to-know) Access Control**

6.2.2.5 The TOE implements two types of discretionary access control mechanism that control the access of subjects to objects that are subject to the administration of access rights:

- a) access control lists for file objects (see 6.1.2.3a))
- b) owner/group/world permissions for non-file objects (see 6.1.2.3b)).

6.2.2.6 Note that owner/group/world permissions also exist for file objects but are expressed as entries in the file object's access control list. For convenience in expressing the requirements, owner/group/world permissions are expressed as applicable to non-file objects only.

#### **Access Control Lists**

6.2.2.7 AC2: The TOE shall ensure that an Access Control List (ACL) is associated with each file object that is subject to DAC.

---

<sup>3</sup> **Application Note:** [EVAL\_CONF] requires the TOE to be configured such that only user-generated passwords are permitted.

<sup>4</sup> **Application Note:** [EVAL\_CONF] requires the TOE to be configured such that all passwords are greater than 7 characters.

- 6.2.2.8 AC3: The ACL shall consist of one or more entries that specify which users or groups of users can access the file object in what access mode.
- 6.2.2.9 AC4: When a file object is created, security related attributes are assigned as follows:
  - a) the object’s owner id shall be the euid associated with the creating process
  - b) the object’s group id shall be the group id of the object’s parent directory
  - c) the creating process must have write permission to the object’s parent directory.
- 6.2.2.10 AC5: The access modes specifiable for ACLs shall include at a minimum read, write, and (for file objects which can be executed) execute access.
- 6.2.2.11 AC6: ACL entries that contain no access modes shall indicate that the specified user(s) or group(s) are specifically excluded from accessing the file object.
- 6.2.2.12 AC7: Whenever a process requests to perform an action on a file object, the ACL for that object shall be checked to determine whether the process can perform the action, as defined in Table 6.1. ACL entries are checked as specified in the rules defined in FDP\_ACF.1.2a). The super-user overrides this discretionary access checking.

<b>Table 6.1 Operations on File Objects and Associated Access Required</b>	
<b>File Object Type</b>	
<b>Action</b>	<b>Access Required</b>
<b>Directories</b>	
Set directory permissions	Owner
Change directory owner or group	
Read entries in a directory	Read
Get directory owner, group and permissions	None
Create a directory entry	Write
Delete a directory entry	
Use directory as a component in a pathname	Execute
<b>Regular files</b>	
Create file	None
Change file owner or group	Superuser
Change file permissions	Owner
Open for reading	Read
Open for writing	Write
Open for reading and writing	Read/Write
Execute program	Execute
<b>Named Pipes (FIFOs)</b>	
Create FIFO	None
Get FIFO Parameters	
Change FIFO owner or group	Superuser
Change FIFO permissions	Owner
Open for reading	Read
Open for writing	Write

<b>Table 6.1 Operations on File Objects and Associated Access Required</b>	
<b>File Object Type</b>	
<b>Action</b>	<b>Access Required</b>
Open for reading and writing	Read/Write
<b>Symbolic Links</b>	
Create a symbolic link Read the contents of a link Get link parameters Use symbolic link as a component in a pathname	None (access controls apply on the target of the link)
<b>Special files</b>	
Create a special file	Super-user only
Get special file parameters	None
Open for reading	Read
Open for writing	Write
Open for reading and writing	Read/Write
Change special file owner or group	Superuser
Change special file permissions	Owner

6.2.2.13 Note that changing a file object’s group may be done by the object’s owner, provided the group the owner tries to change it to is in the list of groups allocated to the owner. This functionality, however, is not claimed, has not been evaluated and is not tested by the TOE’s security test suite.

**Owner/Group/World Permissions**

6.2.2.14 AC8: The ‘owner/group/world permission’ mechanism shall associate with each non-file object that is subject to DAC an owner identification, a group identification, and a set of access permissions. These permissions shall specify the allowable access modes of:

- a) the owner of the object (owner)
- b) any member of the group specified (group)
- c) any user other than the owner or a group member (world).

6.2.2.15 AC9: The access modes specifiable for owner/group/world permissions shall include at a minimum read and write access.

6.2.2.16 AC10: When a non-file object is created security related attributes are assigned as follows:

- a) the object’s owner and creator shall be the euid associated with the creating process
- b) the object’s group shall be the egid associated with the creating process.

6.2.2.17 AC11: Whenever a process requests to perform an action on a non-file object, the owner/group/world access permissions for that object shall be checked in accordance with the rules defined in FDP\_ACF.1.2b) against the user and gids associated with the process to determine whether the process can perform the action, as defined in Table 6.2. The super-user overrides this discretionary access checking.

<b>Table 6.2 Operations on Non-file Objects and Associated Access Required</b>	
<b>Non-file Object Type</b>	
<b>Action</b>	<b>Access Required</b>
<b>Process</b>	
Send a signal to a process Read and write to process address space Set process parameters	Owner
Get process parameters Duplicate process Create new process (i.e. exec)	None
<b>Message queues</b>	
Create message queue	None
Delete message queue Set message queue parameters	Owner and Creator
Get message queue parameters Receive message	Read
Send message	Write
<b>Semaphore sets</b>	
Create semaphore set	None
Delete semaphore set Set semaphore set parameters	Owner and Creator
Get semaphore set parameters Obtain value of a semaphore	Read
Decrement the value of a semaphore Increment the value of a semaphore	Write
<b>Shared memory segments</b>	
Create shared memory segment	None
Delete shared memory segment Set shared memory segment parameters	Owner and Creator
Get shared memory segment parameters Attach to shared memory segment Detach from shared memory segment Read from shared memory segment	Read
Write to shared memory segment	Write
<b>Un-named pipes</b>	
Create unnamed pipes	None
Get un-named pipes parameters Read from un-named pipe Write to un-named pipe	Owner
<b>Pseudo-terminals</b>	
Open a pseudo-terminal for reading Reading from a pseudo-terminal	File read
Open a pseudo-terminal for writing Writing to a pseudo-terminal	File write
Open a pseudo-terminal for reading and writing	File read/write
<b>UNIX Domain Sockets</b>	
Create a socket Bind to a socket	None

<b>Table 6.2 Operations on Non-file Objects and Associated Access Required</b>	
<b>Non-file Object Type</b>	
<b>Action</b>	<b>Access Required</b>
Listen on a socket (Streams only) Accept a socket (Streams only) Read from socket Get socket options Get socket access permissions Get socket owner Set socket options	
Connect to a socket Write to a socket	File write
Change socket access permissions	Owner
Change socket owner or group	Superuser

6.2.2.18 Note that changing a socket’s group may be done by the object’s owner, provided the group the owner tries to change it to is in the list of groups allocated to the owner. This functionality, however, is not claimed, has not been evaluated and is not tested by the TOE’s security test suite.

6.2.2.19 Note also that a socket can be regarded as a file object since its name appears as a directory entry, and can be made subject to ACL’s as a file object can. This functionality also is not claimed, has not been evaluated and is not tested by the TOE’s security test suite.

**Administration**

6.2.2.20 AC12: Only the super-user shall be able to:

- a) create new accounts
- b) delete, disable or enable existing user accounts
- c) assign or modify group affiliations.

**6.2.3 Audit**

**Introduction**

6.2.3.1 All security-relevant operations performed by the TOE are auditable. This includes all operations that make access control decisions, all administrative operations and several other operations.

6.2.3.2 The TOE maintains an audit trail in a standard system file protected by the file-system discretionary access control mechanisms.

6.2.3.3 The super-user is responsible for the following audit-related tasks:

- a) configuring the TOE for auditing, including the options available when the audit file is full
- b) determining what is written to the audit trail
- c) determining what is read from the audit trail.

6.2.3.4 All audit trail maintenance tasks are accomplished through utility programs which are part of the TOE.

6.2.3.5 Where possible, the TOE does not collect all information on every event - additionally the information required for each audit record is reconstructed by the selective reduction program. (Note that the audit collection mechanism does not necessarily have to collect all associated information with each event, but must collect enough data that an audit reduction program can reliably deduce and present the specified information.)

#### **Audit Data**

6.2.3.6 The following paragraphs of this subsection address the minimum set of audit data which the TOE must be capable of collecting. The TOE may also collect additional audit data.

6.2.3.7 AUD1: The TOE shall be able to create, maintain and protect from modification or unauthorised access or destruction<sup>5</sup>, an audit trail of access to the objects it protects.

6.2.3.8 AUD2: The TOE shall be able to:

- a) accept audit data from processes
- b) collect audit data concerning all the events listed in Table 6.3.

6.2.3.9 AUD3: The TOE shall collect sufficient data to allow the generic events and associated information specified in Table 6.3 to be presented to the reviewer of the audit trail.

6.2.3.10 AUD4: The TOE shall ensure that, for all the generic events listed in Table 6.3:

- a) the event type, relevant uid, audit id, date, time, terminal id (if appropriate) and event result (success or failure) is available
- b) any 'additional associated information' as specified in Table 6.3.

6.2.3.11 AUD5: The date and time inserted into audit data shall be reliable<sup>6</sup>.

#### **Selective Collection and Reduction**

6.2.3.12 AUD6: The TOE shall be capable of collecting enough data to satisfy the requirements in Table 6.3 at all times and shall be capable of selectively collecting audit data. At a minimum, it shall be possible to configure the TOE to collect selectively, based on the identity of individuals and the type of audit event.

6.2.3.13 AUD7: The TOE shall provide audit reduction software which permits, at a minimum, the selected retrieval of audit data based on the following:

- a) the identity of individuals
- b) the type of audit event
- c) the identity of objects accessed
- d) time interval.

#### **Local Data Storage**

6.2.3.14 AUD8: The audit data shall be protected so that access to it is limited to the super-user.

---

<sup>5</sup> **Application Note:** The audit daemon can be configured to write audit buffers periodically to disk instead of only when the buffers are full, thereby minimising audit data loss in the event of hardware or power failures. [EVAL\_CONF] describes how the frequency at which audit buffers are written to disk should be selected in accordance with an appropriate business risk management decision which trades-off the residual risk against loss of performance.

<sup>6</sup> **Application Note:** [EVAL\_CONF] advises how often the system clock needs to be re-set in order to maintain acceptable accuracy of the date and time in audit data.



**Audit Trail Exhaustion**

6.2.3.15

AUD9: Should the TOE not be able to write audit data to disk or the disk usage threshold of ninety percent is reached the TOE shall write a message to the audit console daemon log, then shutdown.

<b>Table 6.3 Audit Events and Audit Data</b>	
<b>Generic Event</b>	<b>Additional Associated Information</b>
Start-up and shutdown of audit functions	
Read the audit log	
Modify system audit mask	The new value of the audit mask
Modify user's audit mask	The new value of the audit mask
Modify audit configuration	
Exceeding audit log disk space threshold	
Audit data cannot be written to disk	
Creation and deletion of audit log files	
Execution of abstract machine test	
Changes to system time	
Shutdown	
Identification and authentication attempts	DeviceID
Logout	
Modification of default object security attributes	
Modify user account	Changes made
Enable user account	SubjectID
Disable User Account	SubjectID
Use of super-user role	Role and origin of the request
Process create (start execution)	SubjectID
Process delete (cease execution)	SubjectID
Make object available (open, execute)	SubjectID, objectID, deviceID, access requested
Make object unavailable (close)	SubjectID, objectID, deviceID
Object create	SubjectID, objectID, deviceID
Object delete	SubjectID, objectID, deviceID
Object rename	SubjectID, old objectID, new objectID
Discretionary access changes by a process	SubjectID, objectID, changes made
Change object attributes	ObjectID, changes made
Application audit	Audit data supplied by application

**6.2.4 Object Reuse**

**Introduction**

6.2.4.1

The TOE enforces object reuse constraints on all objects, as well as on process memory and all buffers returned by system calls.

**Object Reuse**

- 6.2.4.2 OR1: When an object is initially assigned, allocated or reallocated to a subject from the TOE’s pool of unused objects, the TOE shall ensure that the object contains no information for which the subject is not authorised.
- 6.2.4.3 OR2: When memory objects are allocated for use by a process at run-time, the memory shall be cleared before the process can read it.
- 6.2.4.4 OR3: Any portion of a file object that has not been previously written to shall either:
  - a) not be readable by any process; or
  - b) be cleared before it can be read.
- 6.2.4.5 OR4: The TOE shall revoke all access rights held by a subject to the information contained within a storage object, prior to initial assignment, allocation or reallocation to another subject.

**6.2.5 Protection Functions**

- 6.2.5.1 The following protection functions are provided by the TOE.
- 6.2.5.2 PF1: The TOE shall use hardware-provided features to prevent itself or its data from being unintentionally or maliciously modified.
- 6.2.5.3 PF2: The TOE shall maintain process isolation through the provision of distinct address spaces under its control.
- 6.2.5.4 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**

This Security Target does not specify any required security mechanisms.

**6.4 Assurance Measures**

The assurance measures adopted to satisfy each of the EAL1 assurance requirements, as defined in [CC] Part 3, Section 6.2.1, Table 6.2, are summarised in Table 6.4.

**Table 6.4 Satisfaction of EAL1 Assurance Requirements by Assurance Measures**

<b>EAL1 Assurance Components</b>	<b>Assurance Measures</b>
ACM_CAP.1 Version numbers	This requirement is met by the TOE.
ADO_IGS.1 Installation, generation, and start-up procedures	This requirement is met by [INSTALL], [EVAL_CONF], [SECURITY], [SYS_ADMIN] and [REL_NOTES].
ADV_FSP.1	This requirement is met by Functional Specification [FS], which

**Table 6.4 Satisfaction of EAL1 Assurance Requirements by Assurance Measures**

<b>EAL1 Assurance Components</b>	<b>Assurance Measures</b>
Informal functional specification	references relevant [REF_PAGES].
ADV_RCR.1 Informal correspondence demonstration	This requirement is met by the [FS].
AGD_ADM.1 Administrator guidance	This requirement is met by [CMND_SHELL], [X_WINDOW], [SYS_ADMIN], [AdvFS], [SECURITY], [EVAL_CONF] and [Man Pages], [REL_NOTES] and [UPDATE].
AGD_USR.1 User guidance	This requirement is met by [CMND_SHELL], [X_WINDOW], [SECURITY], [EVAL_CONF] and [Man Pages].
ATE_IND.1 Independent testing – conformance	Representative platform(s) are provided to enable the evaluators to perform independent functional testing.

## 7 Rationale

### 7.1 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 identical with those specified in [CAPP], except for the environmental assumptions A.PEER and A.CONNECT. Therefore, except for the environmental assumptions A.PEER and A.CONNECT, the security objectives rationale presented in [CAPP] Section 7.1 applies and is not repeated here.

The [CAPP] environmental assumption A.PEER has been replaced for the TOE by a new environmental assumption A.NETWORK. This is to reflect the fact that, whereas [CAPP] is targeted at a TOE that may operate in a distributed local network of identical TOEs, the TOE specified in this Security Target is targeted at a single TOE offering ftp and telnet services, constrained by the TOE's user authentication security functions.

The [CAPP] environmental assumption A.CONNECT has been modified simply to remove the reference to [CAPP] conformance, i.e. to replace 'CAPP conformant TOEs only address' by 'The TOE only addresses'.

A.NETWORK maps to O.INSTALL in the same way that A.PEER maps to O.INSTALL in [CAPP] Section 7.1.3.

A.CONNECT maps to O.PHYSICAL as in [CAPP] Section 7.1.3.

### 7.2 Security Requirements Rationale

#### 7.2.1 Security Functional Requirements Cover Security Objectives

The security functional requirements for the TOE are derived from [CAPP], with no augmentation and with all operations required by [CAPP] carried out (see Section 5.1). The specification of security objectives for the TOE and the environment in Chapter 4 are identical with those specified in [CAPP]. Therefore the rationale for 'complete coverage – objectives' in [CAPP] Section 7.2.2 applies and is not repeated here.

#### 7.2.2 Internal Consistency of Requirements

The security functional requirements for the TOE are derived from [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.

#### 7.2.3 Satisfaction of Dependencies

The security functional requirements for the TOE are derived from [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, apply and are not repeated here.

#### 7.2.4 Justification of Assurance Level

The claimed evaluation assurance level of EAL1 is justified by market requirements, and is appropriate for the type of threats, security objectives and environment claimed.

## 7.3 TOE Summary Specification Rationale

### 7.3.1 Satisfaction of TOE Security Functional Requirements

7.3.1.1 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, AUD2, AUD3
FAU_GEN.1.2	Audit Data Generation	AUD3, AUD4
FAU_GEN.2.1	User Identity Association	I&A6, AUD4
FAU_SAR.1.1	Audit Review	AUD7, AUD8
FAU_SAR.1.2	Audit Review	AUD7
FAU_SAR.2.1	Restricted Audit Review	AUD8
FAU_SAR.3.1	Selectable Audit Review	AUD7
FAU_SEL.1.1	Selective Audit	AUD6
FAU_STG.1.1	Guarantees of Audit Data Availability	AUD8
FAU_STG.1.2	Guarantees of Audit Data Availability	AUD1, AUD8
FAU_STG.3.1	Action in Case of Possible Audit Data Loss	AUD9
'FAU_STG.4.1'	Prevention of Audit Data Loss	AUD9
FDP_ACC.1.1	Discretionary Access Control Policy	AC2, AC7, AC8, AC11
FDP_ACF.1.1	Discretionary Access Control Functions	AC3, AC4, AC5, AC8, AC9, AC10
FDP_ACF.1.2	Discretionary Access Control Functions	AC7, AC11
FDP_ACF.1.3	Discretionary Access Control Functions	AC7, AC11
FDP_ACF.1.4	Discretionary Access Control Functions	AC6
FDP_RIP.2 <sup>1</sup> .1	Object Residual Information Protection	OR1, OR2, OR3, OR4
'FDP_RIP.2 <sup>2</sup> .1' (Note 1)	Subject Residual Information Protection	OR1, OR2, OR3, OR4

**Table 8.1 Mapping of Security Functions to Security Functional Requirements**

Security Functional Requirements		Security Functions
Element	Component Name	
FIA_ATD.1.1	User Attribute Definition	I&A7
FIA_SOS.1.1	Strength of Authentication Data	I&A3, I&A4, I&A16, I&A17
FIA_UAU.1.1	Authentication	I&A1
FIA_UAU.1.2	Authentication	I&A1, I&A8
FIA_UAU.7.1	Protected Authentication Feedback	I&A2
FIA_UID.1.1	Identification	I&A1
FIA_UID.1.2	Identification	I&A1, I&A2
'FIA_USB.1.1' (Note 2)	User-Subject Binding	I&A5, I&A6, I&A8, I&A9
'FIA_USB.1.2' (Note 2)	User-Subject Binding	I&A5, I&A6, I&A8, I&A10
'FIA_USB.1.3' (Note 2)	User-Subject Binding	I&A11, I&A12
FMT_MSA.1.1	Management of Object Security Attributes	AC7, AC11
FMT_MSA.3.1	Static Attribute Initialisation	AC1
FMT_MSA.3.2	Static Attribute Initialisation	AC7, AC11
FMT_MTD.1 <sup>1</sup> .1	Management of the Audit Trail	AUD1, AUD8
FMT_MTD.1 <sup>2</sup> .1	Management of Audited Events	AUD8
FMT_MTD.1 <sup>3</sup> .1	Management of User Attributes	I&A14, AC12
FMT_MTD.1 <sup>4</sup> .1 <sup>1</sup>	Management of Authentication Data	I&A15
FMT_MTD.1 <sup>4</sup> .1 <sup>2</sup>	Management of Authentication Data	I&A13, I&A14
FMT_REV.1 <sup>1</sup> .1	Revocation of User Attributes	I&A14, AC12
FMT_REV.1 <sup>1</sup> .2	Revocation of User Attributes	I&A14, AC12
FMT_REV.1 <sup>2</sup> .1	Revocation of Object Attributes	AC7, AC11
FMT_REV.1 <sup>2</sup> .2	Revocation of Object Attributes	AC7, AC11
FMT_SMR.1.1	Security Management Roles	I&A7, AC7, AC11
FMT_SMR.1.2	Security Management Roles	I&A7, AC7, AC11
FPT_AMT.1.1	Abstract Machine Testing	PF3
FPT_RVM.1.1	Reference Mediation	AC7, AC11
FPT_SEP.1.1	Domain Separation	PF1

**Table 8.1 Mapping of Security Functions to Security Functional Requirements**

Security Functional Requirements		Security Functions
Element	Component Name	
FPT_SEP.1.2	Domain Separation	PF2
FPT_STM.1.1	Reliable Time Stamps	AUD5

7.3.1.2 The following notes are provided to clarify (where necessary) the correspondence between TOE security functional requirements and TOE security functions in Table 8.1:

- a) FIA\_ATD.1.1 requires a minimum set of security attributes belonging to individual users. User group membership in I&A7 includes the user’s login group id specified in **passwd(4)** and supplementary groups mapped to the user in **group(4)**. The user’s Authentication Data is the encrypted password (note that the TOE uses the term authentication data to refer to all user security attributes stored in **prpasswd(4)**). Security-relevant Roles are implicit in the user’s uid (see under FMT\_SMR.1.1 below).
- b) FIA\_SOS.1.1 defines metrics concerning the password strength. The metrics for a single random attempt are met by I&A16 and I&A17 together with the associated application notes, which require users to choose a password of eight or more characters. The metrics for multiple random attempts in one minute are met by the addition of I&A3 together with the associated application note, which ensures the user account is locked after a maximum of five failed authentication attempts. The metric to ensure feedback does not weaken the password strength is met by I&A4
- c) FMT\_SMR.1.1 requires the TOE to maintain certain roles. I&A7 maintains user security attributes, including the user’s user id (uid) – the authorised administrator role is assigned to the super user, which is implicitly defined as a user with a uid of zero. AC7 and AC11 specify that object security attributes may only be changed by a super user or the owner (or creator for non-file objects) of the object. I&A14 specifies that all users shall be able to modify their own authentication data (i.e. passwords)
- d) FMT\_SMR.1.2 is met explicitly, since the role is expressed in terms of the user’s uid in the context of the role being considered.

**7.3.2 Justification of Compliance with Assurance Requirements**

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