Red Hat
Enterprise Linux 3
(running on specified
Dell and Hewlett-Packard hardware)

Security Target

Version 1.7
January 2004

Document Control

DOCUMENT TITLE	Red Hat Enterprise Linux 3 Security Target

Version	Date	Description
1.0	15 August 2003	First release
1.1	1 September 2003	First update
1.2	16 September 2003	Second update
1.3	21 October 2003	Third update
1.4	24 October 2003	Fourth update
1.5	26 November 2003	Fifth update
1.6	15 December 2003	Sixth update
1.7	27 January 2004	Update to address certifier concerns

Acknowledgements

TOE Distributor

The distributor of the TOE is:
Red Hat Inc
1801 Varsity Drive
Raleigh, NC 27606
United States of America

Contact: Donald Fischer (dff@redhat.com)

Evaluation Sponsor

The evaluation sponsor is:
Oracle Corporation
Oracle Parkway
Thames Valley Park
Reading
Berkshire
RG6 1RA
United Kingdom

Contact: seceval_us@oracle.com

ST Preparation

The ST was prepared by:
Syntegra
Sentinel House
Harvest Crescent
Ancells Park
Fleet
Hampshire
GU51 2UZ
United Kingdom

Contact: clef@syntegra.com

Contents

References

[CC] Common Criteria for Information Technology Security Evaluation, Version 2.1, August 1999 (with CCIMB interpretations in place as at 1 July 2003)

[CAPP] Controlled Access Protection Profile, Version 1.d

1.0 Introduction

This section contains document management and overview information necessary to allow the ST to be registered. The ST identification provides the labelling and descriptive information necessary to identify, catalogue, register, and cross-reference the ST. The ST overview summarises the ST in narrative form and provides sufficient information for a potential user to determine whether the TOE is of interest. The conventions section provides an explanation of how this document is organised and the terms section gives a basic definition of terms which are specific to this ST.

1.1 Security Target Identification

Title: Security Target for Red Hat Enterprise Linux 3

Version: 1.7

1.2 Overview

Red Hat Enterprise Linux is a commercially available distribution of the Linux operating system. It is a multi-user, multi-tasking operating system designed for use on Intel x86 platforms.

The TOE is Red Hat Enterprise Linux AS, WS and ES 3.

The TOE supports access controls that are capable of enforcing access limitations on individual users and data objects. It also provides an audit capability that records the security-relevant events that occur within the system.

The TOE provides for a level of protection which is appropriate for an assumed non-hostile and well-managed user community requiring protection against threats of inadvertent or casual attempts to breach the system security. The TOE is not intended to be applicable to circumstances in which protection is required against determined attempts by hostile and well-funded attackers to breach system security. The TOE does not fully address the threats posed by malicious system development or administrative personnel. The TOE is suitable for use in both commercial and government environments.

The TOE is generally applicable to distributed systems, but does not address the security requirements that arise specifically out of the need to distribute the resources within a network.

The security functional requirements used in this ST have followed closely the specification contained in CAPP[1].

1.3 Strength of Environment

The TOE is intended to provide a moderate level of protection for an environment with a low level of risk to the assets. The assurance requirements and the minimum strength of function were chosen to be consistent with that level of risk.

1.4 CC Conformance Claims

This ST is CC Part 2 extended and CC Part 3 conformant. The assurance level is EAL 2 and the minimum strength of function is SOF medium.

1.5 Conventions

This document is organised based on Annex C of Part 1 of the Common Criteria. There are several deviations in the organisation of this ST. Firstly, application notes have

[1] Controlled Access Protection Profile, version 1.d

been integrated with requirements and indicated as notes. Similarly, the rationale has been integrated where appropriate.

For each component, an application note may appear. Application notes provide guidance on how the requirement is applied. Following the application note is rationale for the inclusion of the component in the requirement set.

In the requirement sections, each subsection represents a requirement family or component, and there is a mnemonic in parenthesis. This refers to the requirement section in the CC from which it was derived. Requirement elements have these references included as half-size text at the end of the element. In some places these references indicate a note instead. These notes represent components or elements that do not appear in the CC, and an explanation can be found in section 10 of this ST. The text that appears in the audit event list in 5.1.1.1 is cross-referenced to the functional requirement component in this ST from which that event was derived.

1.6 Terms

This ST uses the following terms, which are described in this section, to aid in the application of the requirements:

- User
- Authorised User
- Authorised Administrator
- Discretionary Access Control (DAC) Policy
- Mediation
- Access
- Authorisation

A **user** is an individual who attempts to invoke a service offered by the TOE.

An **authorised user** is a user who has been properly identified and authenticated. These users are considered to be legitimate users of the TOE.

An **authorised administrator** is an authorised user who has been granted the authority to manage the TOE. These users are expected to use this authority only in the manner prescribed by the guidance given them.

The **Discretionary Access Control Policy**, also referred to as DAC, is the basic policy that a CAPP conformant TOE enforces over users and resources.
Whether a user is granted a requested action is determined by the TOE Security Policy (TSP) which is specified in this profile in the context of Discretionary Access Control (DAC). The DAC policy is the set of rules used to mediate user access to TOE protected objects and can be generally characterised as a policy which requires the TOE to allow authorised users and authorised administrators to control access to objects based on individual user identification. When the DAC policy rules are invoked, the TOE is said to be mediating access to TOE protected objects. However, there may be instances when the DAC policy is not invoked meaning that there may be objects residing in the TOE which are not protected by the TSP. In these instances the TOE is said to not be **mediating** access to a set of objects even though the TOE is executing a (possibly unauthorised) user request.
The DAC policy consists of two types of rules: those that apply to the behaviour of authorised users (termed access rules) and those that apply to the behaviour of authorised administrators (termed authorisation rules). If an authorised user is granted a request to operate on an object, the user is said to have **access** to that object. There are numerous types of access; typical ones include read access and write access which allow the reading and writing of objects respectively. If an authorised administrator is granted a requested service, the user is said to have **authorisation** to the requested service or object. As for access, there are numerous possible authorisations. Typical authorisations include auditor authorisation which allows an

administrator to view audit records and execute audit tools and DAC override authorisation which allows an administrator to override object access controls to administer the system.

1.7 List of Abbreviations

ACL	Access Control List
CAPP	Controlled Access Protection Profile
CC	Common Criteria
CPU	Central Processor Unit
DAC	Discretionary Access Control
DoD	Department of Defense
EAL	Evaluation Assurance Level
GID	Group Identifier
ICMP	Internet Control Message Protocol
IPX	Internetwork Packet Exchange
PP	Protection Profile
SOF	Strength of Function
SFR	Security Functional Requirement
SSH	Secure Shell
ST	Security Target
TCP	Transport Control Protocol
TOE	Target of Evaluation
TSC	TOE Scope of Control
TSF	TOE Security Functions
TSP	TOE Security Policy
UID	User Identifier
UDP	User Datagram Protocol
VFS	Virtual File System

2.0 TOE Description

2.1 Introduction

Linux is a free computer operating system that was created in 1991 by Linus Torvalds, based on POSIX standards, and has grown through contributions from software developers all over the world.

Red Hat Enterprise Linux is a commercially supported distribution of the free Linux operating system that is easier to install and operate. Red Hat Enterprise Linux is designed for mission-critical enterprise computing, with support for the largest X86-compatible servers, used in departmental and datacentre server deployments.

The TOE assumes that responsibility for the safeguarding of the data protected by the TOEs security functions (TSF) can be delegated to the TOE users. All data are under the control of the TOE. The data are stored in objects, and the TSF can associate with each controlled object a description of the access rights to that object.

2.2 TOE Architecture

Red Hat Enterprise Linux (also referred to in this document as Linux) provides a multi-user, multi-tasking environment. The operating system may be viewed as a series of layers. At the lowest layer, the Linux kernel interacts with the hardware platform, providing a common set of services to application programs. These services include managing system memory, sharing access to the system processor(s), and opening and closing devices. In addition, the operating system provides other basic services, including:

- File systems organised within a hierarchy of directories;
- Device drivers providing interfaces to hardware devices
- User interfaces to run programs and access the file system. Linux includes both graphical interfaces (GNOME and KDE) and shell command interpreters (e.g. bash). Note that the graphical interfaces are excluded from the evaluated configuration.
- System utilities, including processes to mount file systems, operate networks and run scheduled tasks.

The most important distinction here is between the kernel and everything else. The kernel operates in the processor's privileged mode, known as kernel mode, with full access to all resources of the computer. Any operating system support code that does not need to run in kernel mode is contained in the system libraries.

The operating system interface visible to running applications is implemented by calls to system libraries, which in turn call operating system services as necessary, rather than by direct calls to operating system services. All functions necessary to support UNIX or POSIX applications are implemented by the system libraries.

Linux also includes a large number of programs that run in user mode, both system utilities and user utilities. Some of the system utilities may be invoked just once, to initialise and configure some aspect of the system, whereas others (daemons) may run permanently (e.g. to accept login requests or update log files). User utilities are provided to perform everyday tasks such as listing directories, moving files, or more complex operations such as text editing.

Kernel Modules

The kernel is implemented as a series of modules that can be loaded and unloaded on demand to run in privileged mode on the processor. These modules will include, for example, device drivers.

Support for module management comprises:

a) Module management that allows modules to be loaded into memory and to communicate with other parts of the kernel;

b) Driver registration that maintains information on what drivers are available;

c) A mechanism for resolving conflicts that allows device drivers to reserve access to hardware resources.

Process Management

A process provides context for the servicing of user activity requests. The creation of a process is distinct from running a program, in that a program may itself create new processes to run in, or a new program can be executed within an existing process.

Each process has a unique identifier, and has an associated User Identifier (UID), and one or more Group Identifiers (GID). These identifiers determine the rights (privileges) of the process to access system resources and files.

A process is created by means of the *fork* command by another process, and inherits the characteristics of its parent. Any two processes can have their own independent address spaces, but may also share address space, application data structures and open files. In this latter case, whenever one of them modifies a shared resource, the other immediately sees the change. Thus, Linux may implement a multi-threaded application by associating a process with each thread. Each process may be scheduled independently by the kernel.

Scheduling

Linux provides scheduling services in order to allocate CPU time to different tasks. These tasks include both user and kernel processes. Scheduling not only manages the allocation of processor time, but also ensures that access to shared data structures by kernel tasks is not disrupted. The scheduler also handles the allocation of tasks to multiple processors.

Memory Management

Memory management services are provided to handle the allocation of physical memory, either as pages, groups of pages or small blocks; and for allocation of virtual memory, mapped into the address space of running processes. The Virtual Memory System maintains the address space visible to each process. It creates pages of virtual memory on demand, and manages the loading of those pages from disk, and their swapping back out to disk as required.

File Systems

Linux files can be anything that is capable of handling the input and output of a stream of data. In addition to stored data objects, files include such things as directories, device drivers and network connections. Implementation details of individual file types are managed by the Virtual File System (VFS).

The operating system kernel maintains a single directory hierarchy of files (a file system) for each disk device mounted as a file system, and for each networked file system.

The VFS is also used to store data associated with processes (the proc file system). Each sub-directory corresponds to an active process on the current system, and additional directories and text files are used to store information about the kernel and loaded drivers. This enables programs to access this information directly in an unprivileged mode.

Input/Output

All device drivers in Linux appear as normal files, and are of three types:

a) Block devices – allowing random access to independent, fixed-size blocks of data e.g. for hard disks;

b) Character devices – most other devices that interact by a stream of characters e.g. mouse, tape;

c) Network devices – used by the kernel's networking subsystem, and accessed only indirectly by users.

Interprocess Communication

For interprocess communication Linux implements Unix signals and semaphores. It also uses wait queues for communication and scheduling between kernel mode processes.

Network Structure

All networking requests are handled by the socket interface. This interface is used in Linux to access all the protocols supported by the system. Below this layer support is provided for the Internet protocol suite. The IP protocol performs routing between hosts on a network. On top of this protocol UDP, TCP and ICMP are supplied.

2.3 Security services provided by the TOE

Identification/Authentication

All individual users are assigned a unique identifier. This identifier supports individual accountability. The TSF authenticates the claimed identity of the user before allowing the user to perform any actions that require TSF mediation, other than actions that aid an authorised user in gaining access to the TOE. Authentication is achieved by means of a password.

Access control

Access control is applied to system objects, including files and shared memory. Every object on a server has a single UID and a single GID associated with it. User processes also have a single UID, but may have more than one GID. Access rights are determined by comparing the UID and GID of the process with those of the object.

Audit

The TOE can be configured to record security relevant events in various audit logs that can be subsequently analysed and displayed by an administrator.

Object Reuse

The TSF ensures that any previous data associated with a storage object is made unavailable on reallocation of the storage object to another user.

Process separation

The TSF maintains separation between the operating system kernel and user processes, and between each user process.

2.4 Evaluated Configuration

The TOE covers the following products, built around a common core:

- Red Hat Enterprise Linux AS 3 – supporting large commodity-architecture servers, for large departmental and datacentre server deployments;

- Red Hat Enterprise Linux ES 3 – suitable for medium scale departmental deployments;

- Red Hat Enterprise Linux WS 3 – the workstation product, suitable for software development or client applications.

The TOE is evaluated on the following hardware platforms:

HP D530	(Red Hat Enterprise Linux WS)
HP Proliant ML570	(Red Hat Enterprise Linux ES and AS)
Dell Precision 650	(Red Hat Enterprise Linux WS)
Dell PE 2650	(Red Hat Enterprise Linux ES)
Dell PE 6650 4 Processor	(Red Hat Enterprise Linux AS)

The following features are excluded from the scope of the TOE, and it is assumed that they are not used:

- Apache Web Server
- Kerberos
- Crypto IP Encapsulation
- Nmap
- LILO
- Network File System (NFS)
- Domain Naming System (DNS)
- Dynamic Host Configuration protocol (DHCP)
- Network Information System (NIS)
- Automatic Updating using Red Hat Up2date
- X-Windows Graphical Interface
- Support for AppleTalk
- Support for IPX
- Red Hat Cluster Manager

3.0 Security Environment

3.1 Threats

This ST has derived all security objectives from the statement of Organisational Security Policy found in the following section. Therefore, there is no statement of the explicit threats countered by the TOE.

3.2 Organisational Security Policies

An Organisational Security Policy is a set of rules or procedures imposed by an organisation upon its operations to protect its sensitive data. The organisational security policies described below apply to many DoD and non-DoD environments.

P.AUTHORISED_USERS
Only those users who have been authorised 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 authorised 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.

3.3 Security Usage Assumptions

This section describes the security aspects of the environment in which the TOE is intended to be used. This includes information about the physical, personnel, and connectivity aspects of the environment.

The TOE is assured to provide effective security measures in a co-operative non-hostile environment only if it is installed, managed, and used correctly. The operational environment must be managed in accordance with assurance requirements documentation for delivery, operation, and user/administrator guidance. The following specific conditions are assumed to exist in an environment where the TOE is employed.

3.3.1 Physical Assumptions

The TOE is intended for application in user areas that have physical control and monitoring. It is assumed that the following physical conditions will exist:

A.LOCATE
The processing resources of the TOE will be located within controlled access facilities that will prevent unauthorised physical access.

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

3.3.2 Personnel Assumptions

It is assumed that the following personnel conditions will exist:

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

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

A.COOP
Authorised users possess the necessary authorisation to access at least some of the information managed by the TOE and are expected to act in a co-operating manner in a benign environment.

3.3.3 Connectivity Assumptions

This ST contains no explicit network or distributed system requirements. However, it is assumed that the following connectivity conditions exist:

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. There are no security requirements that 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. The TOE only addresses security concerns related to the manipulation of the TOE through its authorised access points. Internal communication paths to access points such as terminals are assumed to be adequately protected.

4.0 Security Objectives

This section defines the security objectives of the TSF and its supporting environment. Security objectives, categorised as either IT security objectives or non-IT security objectives, reflect the stated intent to counter identified threats and/or comply with any organisational security policies identified. All of the identified threats and organisational policies are addressed under one of the categories below.

4.1 IT Security Objectives

The following are the TOE IT security objectives:

O.AUTHORIZATION
The TSF must ensure that only authorised 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 authorised users to specify which resources may be accessed by which users.

O.AUDITING
The TSF must record specified security relevant actions of users of the TOE. The TSF must present this information to authorised 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 authorised administrators that are responsible for the management of TOE security.

O.ENFORCEMENT
The TSF must be designed and implemented in a manner that ensures that the organisational policies are enforced in the target environment.

4.2 Non-IT Security Objectives

The TOE is assumed to be complete and self-contained and, as such, is not dependent upon any other products to perform properly. However, certain objectives with respect to the general operating environment must be met. The following are the TOE non-IT security objectives:

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 that 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 that maintains IT security objectives.

5.0 Security Functional Requirements

This chapter defines the security functional requirements for the TOE. Most of the security functional requirement components in this ST were drawn from Part 2 of the CC. Some functional requirements are extensions to those found in the CC.

CC defined operations for assignment, selection, and refinement have been used to tailor the requirements to the level of detail necessary to meet the stated security objectives. These operations are indicated through the use of underlined (assignments and selections) and italicised (refinements) text.

5.1 Security Audit (FAU)

5.1.1 Audit Data Generation (FAU_GEN.1)

5.1.1.1 The TSF shall be able to generate an audit record of the auditable events *listed in column "Event" of Table 1 (Auditable Events).*
FAU_GEN.1.1 / NOTE 3

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

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

b) The additional information specified in the "Details" column of Table 1 (Auditable Events).

Rationale: This component supports O.AUDITING by specifying the detailed, security-relevant events and data that the audit mechanism must be capable of generating and recording.

Table 1: Auditable Events

Section	Component	Event	Details
5.1.1	FAU_GEN.1	Start-up and shutdown of the audit functions.	
5.1.2	FAU_GEN.2	None	
5.1.3	FAU_SAR.1	None.	
5.1.4	FAU_SAR.2	None.	
5.1.5	FAU_SAR.3	None	
5.1.6	FAU_STG.1	None	
5.2.1	FDP_ACC.1	None	
5.2.2	FDP_ACF.1	None	
5.2.3	FDP_RIP.2	None	
5.2.4	Note 1	None	
5.3.1	FIA_ATD.1	None	
5.3.2	FIA_SOS.1	Rejection by the TSF of any tested secret.	
5.3.3	FIA_UAU.2	All use of the authentication	

		mechanism.	
5.3.4	FIA_UAU.7	None	
5.3.5	FIA_UID.2	All use of the user identification mechanism, including the identity provided during successful attempts.	The origin of the attempt (e.g. terminal identification.)
5.3.6	FIA_USB.1	None	
5.4.1	FMT_MSA.1	Modifications to the values of user security attributes.	
5.4.2	FMT_MSA.3	None	
5.4.3	FMT_MTD.1	None	
5.4.4	FMT_MTD.1	None	
5.4.5	FMT_MTD.1	None	
5.4.6	FMT_MTD.1	None	
5.4.7	FMT_REV.1	All attempts to revoke security attributes associated with a user.	
5.4.8	FMT_REV.1	None	
5.4.9	FMT_SMF.1	None	
5.4.10	FMT_SMR.1	Modifications to the group of users that are part of a role.	
5.4.10	FMT_SMR.1	None	
5.5.1	FPT_AMT.1	None	
5.5.2	FPT_RVM.1	None	
5.5.3	FPT_SEP.1	None	
5.5.4	FPT_STM.1	None	

5.1.2 User Identity Association (FAU_GEN.2)

5.1.2.1 The TSF shall be able to associate each auditable event with the identity of the user that caused the event. FAU_GEN.2.1

Application Note: There are some auditable events that may not be associated with a user, such as failed login attempts. It is acceptable that such events do not include a user identity. In the case of failed login attempts it is also acceptable not to record the attempted identity in cases where that attempted identity could be misdirected authentication data; for example when the user may have been out of sync and typed a password in place of a user identifier.

Rationale: O.AUDITING calls for individual accountability (i.e., “TOE users”) whenever security-relevant actions occur. This component requires every auditable event to be associated with an individual user.

5.1.3 Audit Review (FAU_SAR.1)

5.1.3.1 The TSF shall provide authorised administrators with the capability to read all audit information from the audit records: FAU_SAR.1.1

5.1.3.2 The TSF shall provide the audit records in a manner suitable for the user to interpret the information. FAU_SAR.1.2

Application Note: The minimum information that must be provided is the same as is required to be recorded in 5.1.1.2. The intent of this requirement is that there should exist a tool for administrator to be able to access the audit trail in order to assess it. Exactly what manner is provided is an implementation decision, but it needs to be done in a way that allows the administrator to make effective use of the information presented. This requirement is closely tied to 5.1.5 and 5.1.6. It is expected that a single tool will exist within the TSF that will satisfy all of these requirements.

Rationale: This component supports the O.AUDITING and O.MANAGE objectives by providing the administrator with the ability to assess the accountability information accumulated by the TOE.

5.1.4 Restricted Audit Review (FAU_SAR.2)

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

Application Note: By default, authorised administrators are considered to have been granted read access to the audit records. The TSF provides a mechanism that allows other users to also read audit records.

Rationale: This component supports the O.AUDITING objective by protecting the audit trail from unauthorised access.

5.1.5 Selectable Audit Review (FAU_SAR.3)

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

a) User identity;

b) Date/time;

c) Event severity;

d) Type of event.

Rationale: This component supports both the O.AUDITING and O.MANAGE objectives, by providing a means for the administrator to assess the accountability information associated with an individual user.

5.1.6 Protected Audit Trail Storage (FAU_STG.1)

5.1.6.1 The TSF shall protect the stored audit records from unauthorised deletion. FAU_STG.1.1

5.1.6.2 The TSF shall be able to prevent unauthorised modifications to the audit records in the audit trail. FAU_STG.1.2

Application Note: In order to reduce the performance impact of audit generation, audit records will be temporarily buffered in memory before they are written to disk. In these cases, it is likely that some of these records will be lost if the operation of the TOE is interrupted by hardware or power failures.

Rationale: This component supports the O.AUDITING objective by protecting the audit trail from tampering, via deletion or modification of records in it. Further it ensures that it is as complete as possible.

5.2 User Data Protection (FDP)

5.2.1 Discretionary Access Control Policy (FDP_ACC.1)

5.2.1.1 The TSF shall enforce the Discretionary Access Control Policy on processes acting on the behalf of users, files, and all operations among subjects and objects covered by the *DAC policy*. FDP_ACC.1.1

Rationale: This component supports the O.DISCRETIONARY_ACCESS objective by specifying the scope of control for the DAC policy.

5.2.2 Discretionary Access Control Functions (FDP_ACF.1)

5.2.2.1 The TSF shall enforce the Discretionary Access Control Policy to objects based on the following: FDP_ACF.1.1

Subjects/Objects	**SFP related security attributes**
User (subject)	User identity, group membership
Process (subject)	Real user identity, effective user identity
File (object)	Owner, group, ACL

5.2.2.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

Object access is granted if at least one of the following conditions is true:

a) The subject is the owner of the object;

b) The owner or administrator has granted access to members of the object's group and the subject is a member of the same group as the object;

c) The owner or administrator has granted access to all subjects.

Access will be for read, write and/or execute, as set by the owner or administrator. FDP_ACF.1.2

5.2.2.3 The TSF shall explicitly authorise access of subjects to objects based in the following additional rules:

a) If the subject is an administrator. FDP_ACF.1.3

5.2.2.4 The TSF shall explicitly deny access of subjects to objects based on the following rules

a) None. FDP_ACF.1.4

Rationale: This component supports the O.DISCRETIONARY_ACCESS objective by defining the rules that will be enforced by the TSF.

5.2.3 Full *Object* Residual Information Protection (FDP_RIP.2)

5.2.3.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the allocation of the resource to all objects. FDP_RIP.2.1

Application Note: This requirement applies to all resources governed by or used by the TSF; it includes resources used to store data and attributes. It also includes the encrypted representation of information.

Rationale: This component supports the O.RESIDUAL_INFORMATION objective.

5.2.4 Full *Subject* Residual Information Protection (FDP_RIP.3 [See Note 1])

5.2.4.1 *The TSF shall ensure that any previous information content of a resource is made unavailable upon the allocation of the resource to all subjects.* FDP_RIP.3.1

Application Note: This requirement applies to all resources governed by or used by the TSF; it includes resources used to store data and attributes. It also includes the encrypted representation of information.

Rationale: This component supports the O.RESIDUAL_INFORMATION objective.

5.3 Identification and Authentication (FIA)

5.3.1 User Attribute Definition (FIA_ATD.1)

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

a) User identifier;

b) Group memberships;

c) Authentication data;

d) Security-relevant roles.

Application Note: The specified attributes are those that are required by the TSF to enforce the DAC policy, the generation of audit records, and proper identification and authentication of users. The user identity must be uniquely associated with a single individual user.
Note that the attribute security-relevant roles is managed as a set of group memberships. Item d) is included here for consistency with the wording in CAPP.

Rationale: This component supports the O.AUTHORIZATION and O.DISCRETIONARY_ACCESS objectives by providing the TSF with the information about users needed to enforce the TSP.

5.3.2 Specification of Secrets (FIA_SOS.1)

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

a) For each attempt to use the authentication mechanism, the probability that a random attempt will succeed is less than one in 1,000,000;

b) For multiple attempts to use the authentication mechanism during a one minute period, the probability that a random attempt during that minute will succeed is less than one in 100,000; and

c) Any feedback given during an attempt to use the authentication mechanism will not reduce the probability below the above metrics.

Rationale: This component supports the O.AUTHORIZATION objective by providing an authentication mechanism with a reasonable degree of certainty that only authorised users may access the TOE.

5.3.3 User Authentication Before Any Action (FIA_UAU.2)

5.3.3.1 The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on the behalf of that user. FIA_UAU.2.1

Rationale: This component supports the O.AUTHORIZATION objective by specifying what actions unauthenticated users may perform.

5.3.4 Protected Authentication Feedback (FIA_UAU.7)

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

Application Note: Obscured feedback implies the TSF does not produce a visible display of any authentication data entered by a user, such as through a keyboard (e.g., echo the password on the terminal). It is acceptable that some indication of progress be returned instead, such as a period returned for each character sent. Some forms of input may contain human-readable user passwords.

Rationale: This component supports the O.AUTHORIZATION objective. Individual accountability cannot be maintained if the individual's authentication data, in any form, is compromised.

5.3.5 User Identification Before Any Action (FIA_UID.2)

5.3.5.1 The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on the behalf of that user. FIA_UID.2.1

Rationale: This component supports the O.AUTHORIZATION objective by specifying what actions unidentified users may perform.

5.3.6 User-Subject Binding (FIA_USB.1)

5.3.6.1 The TSF shall associate the *following* user security attributes with subjects acting on the behalf of that user: FIA_USB.1.1 / NOTE 2

a) The user identity that 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.

5.3.6.2 *The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of a user*: NOTE 2

a) upon successful identification and authentication the user ID shall be that specified by the User Identity attribute held by the TSF for the user;

b) *upon successful identification and authentication the group IDs shall be that specified by the Group Membership attribute held by the TSF for the user*.

5.3.6.3 *The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of a user*: NOTE 2

a) The user ID associated with a subject can be changed to another user's identity by a command, provided that the original user ID was 0, or successful authentication as the new user ID has been achieved;

b) When executing a file that has the Set UID permission bit set the effective user ID associated with the subject shall be changed to that of the owner of the file;

c) *When executing a file that has the Set GID permission bit set the effective group ID associated with the subject shall be changed to that of the group attribute of the file*.

Rationale: This component supports the O.DISCRETIONARY_ACCESS and O.AUDITING objectives by binding user identities to subjects acting on their behalf.

5.4 Security Management (FMT)

5.4.1 Management of *Object* Security Attributes (FMT_MSA.1)

5.4.1.1 The TSF shall enforce the Discretionary Access Control Policy to restrict the ability to modify the access control attributes associated with a named object to the owner of the object or an authorised administrator. FMT_MSA.1.1

Rationale: This component supports the O.DISCRETIONARY_ACCESS objective by providing the means by which the security attributes of objects are managed by a site.

5.4.2 Static Attribute Initialisation (FMT_MSA.3)

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

5.4.2.2 The TSF shall allow the owner of the object or an authorised administrator to specify alternative initial values to override the default values when an object or information is created. FMT_MSA.3.2

Application Note: The TOE must provide protection by default for all objects at creation time. This is done through the enforcing of a restrictive default access control on newly created objects There shall be no window of vulnerability through which unauthorised access may be gained to newly created objects.

Rationale: This component supports the O.DISCRETIONARY_ACCESS objective by requiring that objects are properly protected starting from the instant that they are created.

5.4.3 Management of TSF Data - Audit Trail (FMT_MTD.1)

5.4.3.1 The TSF shall restrict the ability to create, delete, and clear the audit trail to authorised administrators. FMT_MTD.1.1

Rationale: The component supports the O.AUDITING and O.MANAGE objectives by ensuring that the accountability information is not compromised by destruction of the audit trail.

5.4.4 Management of TSF Data - Audited Events (FMT_MTD.1)

5.4.4.1 The TSF shall restrict the ability to modify or observe the set of audited events to authorised administrators. FMT_MTD.1.1

Application Note: The set of “audited events” are the subset of auditable events that will be audited by the TSF. The term set is used loosely here and refers to the total collection of possible ways to control which audit records get generated; this could be by type of record, identity of user, identity of object, etc. It is an important aspect of audit that users not be able to detect which of their actions are audited, and therefore must not have control over or knowledge of the selection of an event for auditing.

Rationale: This component supports the O.AUDITING and O.MANAGE objectives by providing the administrator with the ability to control the degree to which accountability is generated.

5.4.5 Management of TSF Data - User Attributes (FMT_MTD.1)

5.4.5.1 The TSF shall restrict the ability to initialise and modify the user security attributes, other than authentication data to authorised administrators. FMT_MTD.1.1

Application Note: This component only applies to security attributes that are used to maintain the TSP.

Rationale: This component supports the O.MANAGE objective by providing the administrator with the means to manage who are authorised users and what attributes are associated with each user.

5.4.6 Management of TSF Data - Authentication Data (FMT_MTD.1)

5.4.6.1 The TSF shall restrict the ability to initialise the authentication data to authorised administrators. FMT_MTD.1.1

5.4.6.2 The TSF shall restrict the ability to modify the authentication data to the following: FMT_MTD.1.1

a) authorised administrators; and

b) users authorised to modify their own authentication data.

Application Note: User authentication data refers to information that users must provide to authenticate themselves to the TSF. Examples include passwords, personal identification numbers, and fingerprint profiles. User authentication data does not include the users identity.
This component does not require that any user be authorised to modify their own authentication information; it only states that it is permissible. It is not necessary that requests to modify authentication data require reauthentication of the requester's identity at the time of the request.

Rationale: This component supports the O.AUTHORIZATION and O.MANAGE objectives by ensuring integrity and confidentiality of authentication data.

5.4.7 Revocation - User Attributes (FMT_REV.1)(U)

5.4.7.1 The TSF shall restrict the ability to revoke security attributes associated with the users within the TSC to authorised administrators. FMT_REV.1.1

5.4.7.2 The TSF shall enforce the rules: FMT_REV.1.2

a) The immediate revocation of security-relevant authorisations.

Application Note: Many security-relevant authorisations could have serious consequences if misused, so an immediate revocation method must exist. Normally revocation does not take effect until the user logs off and logs back on. The method for immediate revocation would be to edit the trusted users profile and "force" the trusted user to log off.

Rationale: This component supports the O.MANAGE objective by controlling access to data and functions that are not generally available to all users.

5.4.8 Revocation - Object Attributes (FMT_REV.1)(O)

5.4.8.1 The TSF shall restrict the ability to revoke security attributes associated with the objects within the TSC to users authorised to modify the security attributes by the Discretionary Access Control policy. FMT_REV.1.1

5.4.8.2 The TSF shall enforce the rules: FMT_REV.1.2

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

Application Note: The TOE operates delayed revocation (e.g. it does not revoke access to already opened files). The DAC access rights are considered to have been revoked when all subsequent access control decisions by the TSF use the new access control information. It is not required by [CAPP] that every operation on an object make an explicit access control decision as long as a previous access control decision was made to permit that operation.

Rationale: This component supports the O.DISCRETIONARY_ACCESS objective by providing that specified access control attributes are enforced at some fixed point in time.

5.4.9 Specification of Management Functions (FMT_SMF.1)

5.4.9.1 The TSF shall be capable of performing the following security management functions: FMT_SMF.1.1

a) modify the access control attributes associated with a named object;

b) create delete and clear the audit trail;

c) modify and observe the set of audited events;

d) initialise and modify the user security attributes that are used to maintain the TSP;

e) initialise and modify the authentication data

f) modify the behaviour of the authentication functions.

Rationale: This component explicitly states the management functions associated with FMT_MSA.1 and FMT_MTD.1. It is not used in CAPP, but was introduced by CCIMB Interpretation 65, and meets new dependencies identified for those components.

5.4.10 Security Management Roles (FMT_SMR.1)

5.4.10.1 The TSF shall maintain the roles: FMT_SMR.1.1

a) authorised administrator (root user);

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

c) users authorised to modify their own authentication data.

5.4.10.2 The TSF shall be able to associate users with roles. FMT_SMR.1.2

Application Note: The TOE supports a number of other distinct administrative roles by default, and also permits others to be defined by means of group membership, but these are not required by [CAPP] and are not addressed by this ST.

Rationale: This component supports the O.MANAGE objective.

5.5 Protection of the TOE Security Functions (FPT)

5.5.1 Abstract Machine Testing (FPT_AMT.1)

5.5.1.1 The TSF shall run a suite of tests at the request of an authorised administrator[2] to demonstrate the correct operation of the security assumptions provided by the abstract machine that underlies the TSF. FPT_AMT.1.1

Application Note: This component refers to the proper operation of the hardware platform on which a TOE is running. The test suite covers only aspects of the hardware on which the TSF relies to implement required functions, including domain separation.

Rationale: This component supports the O.ENFORCEMENT objective by demonstrating that the underlying mechanisms are working as expected.

[2] CAPP offers the choice to schedule these tests on start-up, periodically or at the request of an authorised administrator

5.5.2 Reference Mediation (FPT_RVM.1)

5.5.2.1 The TSF shall ensure that the TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed. FPT_RVM.1.1

Application Note: This element does not imply that there must be a reference monitor. Rather this requires that the TSF validates all actions between subjects and objects that require policy enforcement.

Rationale: This component supports O.ENFORCEMENT objective by ensuring that the TSP is not being bypassed.

5.5.3 TSF Domain Separation (FPT_SEP.1)

5.5.3.1 The TSF shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects. FPT_SEP.1.1

5.5.3.2 The TSF shall enforce separation between the security domains of subjects in the TSC. FPT_SEP.1.2

Application Note: This component does not imply a particular implementation of a TOE. The implementation needs to exhibit properties that the code and the data upon which TSF relies are not alterable in ways that would compromise the TSF and that observation of TSF data would not result in failure of the TSF to perform its job.

Rationale: This component supports O.ENFORCEMENT objectives by ensuring that a TSF exists within the TOE and that it can reliably carry out its functions.

5.5.4 Reliable Time Stamps (FPT_STM.1)

5.5.4.1 The TSF shall be able to provide reliable time stamps for its own use. FPT_STM.1.1

Application Note: The generation of audit records depends on having a correct date and time. The word “reliable” in the above requirement means that the order of occurrence of auditable events is preserved.

Rationale: This component supports the O.AUDITING objective by ensuring that accountability information is accurate.

6.0 Assurance Requirements

This chapter defines the assurance requirements for the TOE. Assurance requirement components are Evaluation Assurance Level (EAL) 2, with no augmentation, from part 3 of the CC.

Authorisation Controls (ACM_CAP.2)

Delivery Procedures (ADO_DEL.1)

Installation, Generation, and Start-up Procedures (ADO_IGS.1)

Functional Specification (ADV_FSP.1)

High-Level Design (ADV_HLD.1)

Correspondence Demonstration (ADV_RCR.1)

Administrator Guidance (AGD_ADM.1)

User Guidance (AGD_USR.1)

Coverage (ATE_COV.1)

Functional Testing (ATE_FUN.1)

Independent Testing (ATE_IND.2)

Strength of TOE Security Function Evaluation (AVA_SOF.1)

Developer Vulnerability Analysis (AVA_VLA.1)

7.0 PP Claims

There are no specific PP claims. However [CAPP] has been used as a model for this ST.

8.0 TOE Summary Specification

8.1 Security Functions

Red Hat Enterprise Linux provides security functions in the following areas:

- Identification/Authentication
- Discretionary Access Control
- Audit
- Object Reuse
- Process Separation
- Self Testing

8.1.1 Identification/Authentication

User Accounts

Users can either be individuals (accounts tied to a physical user) or logical (accounts that exist for applications to perform specific tasks). Both types of users have a User ID (UID), a Group ID (GID), and a password [IA.1].

Groups are logical expressions of organisation. Groups allow users to be associated, and to receive common permissions to read, write or execute a given file [IA.2].

User identification and group membership attributes can be modified only by an administrator [IA.3]

Authentication

The TOE will authenticate a user's identity before access is granted to any resources under control of the TOE [IA.4][3]

Successful authentication requires all of the following to be true:

a) the user has entered a valid user name;

b) the password entered by the user, and encrypted by the TOE, is identical to the encrypted password stored by the TOE for that user;

c) the user account is not locked;

d) the password has not expired [IA.5].

A user account will be locked by the TOE if any of the following are true:

a) the number of consecutive failed login attempts to that user account exceeds the maximum permitted;

b) the user account has been explicitly locked by an administrator [IA.6].

An administrator can define the following restrictions on the authentication process:

a) the maximum number of days for which a password is valid;

b) the minimum acceptable password length;

c) the maximum number of consecutive failed login attempts before a user account is locked;

[3] Realised by a probabilistic/permutational mechanism with strength SOF-Medium

d) whether a user can modify their password;

e) the set of unacceptable passwords (through use of the CrackLib proactive password checking library) [IA.7].

Shadow passwords

Encrypted passwords are held in a shadow password file that is readable only by an administrator [IA.8].

User identification

When a User ID and password are entered the password is encrypted and compared with the representation held in the shadow password file for that User ID. Access is granted only if the password matches [IA.9].

Changing user identity

A user may change identity by means of the su command. By this means they may adopt, following successful authentication, the real and effective User id and Group id of another user [IA.10]. The most common use of this function would be to adopt the root identity in order to carry out administrative tasks. The root administrator account may also be accessed by means of login, but only at the directly attached console.[IA.11].

Password selection

Unless prevented by an administrator, users are permitted to change their password through use of the *passwd* command. Users are required to re-authenticate before being permitted to change their passwords, whereas authorised administrators are not [IA.12].

Initial passwords are set only by an administrator [IA.13]

Protected feedback

The TSF will not echo user passwords to the display when they are entered in the password field. No indication is given on the display that any character has been entered [IA.14].

Password encryption

The TSF will encrypt passwords using the MD5 message-digest algorithm [IA.15][4].

Forced logoff

An administrator can force the log off of a user by killing processes associated with that user [IA.16].

8.1.2 Discretionary Access Control

The TSF mediates access between subjects (processes), acting on behalf of users, and objects within its scope of control. These objects are:

a) files;

b) directories;

c) volumes;

[4] The correct implementation of MD5 was not verified during the evaluation.

d) sockets;

e) named pipes

f) shared memory;

g) message queues;

h) semaphores.
[DAC.1].

The kernel associates two User ids with a process, independent of the process id: the real user id and the effective user id. The real user id identifies the user responsible for running the process. The effective user id is used to assign ownership of newly created files or to check file access permissions. This also applies in the case of group id [DAC.2].

For access control of data objects (files), users may be either:

- The owner of the file;
- Users belonging to the same group as the file (not including the owner);
- Other users.

There are three types of access rights (read, write, execute) for each of the above categories of user. These access rights can be modified by the file owner. The set of access rights associated with a file therefore consists of nine binary flags. A user is grated access to a file in the requested mode (read, write or execute) only if access is permitted by the flag settings [DAC.3].

Three additional flags define the file mode. These are suid (Set User ID), sgid (Set Group ID) and sticky. When applied to executable files (no effect on non-executable files) these flags have the following meanings:

Suid A process normally retains the user ID of the process owner when executing a file. If this flag is set the process takes on the user ID of the file owner.

Sgid A process normally retains the group ID of the process owner when executing a file. If this flag is set the process takes on the ID of the file group.

Sticky The kernel is requested to keep the program in memory following termination]

When the sticky bit is set on a directory, files in that directory may be unlinked or renamed only by root or their owner. Without the sticky bit, anyone able to write to the directory can delete or rename files. The sticky bit is commonly used on directories such as /tmp that are world writable. [DAC.4].

For a given file object access permissions are defined in an Access Control List (ACL). The ACL identifies access permissions for the owner of the file, members of the group that has the same Group ID as the file, and a default for other users. In addition it may include specific permission bit sets for other identified individuals, or members of other identified groups [DAC.5].

The TOE provides the ability for an administrator or user to specify default access permissions for the creation of data objects [DAC.6].

A default ACL may be associated with a directory. The default ACL specifies initial default access rights for files created in that directory. If the new file created is a directory it will inherit the default ACL of its parent directory [DAC.7].

An administrator (root) can access any file, and can change file ownership or access permissions [DAC.8].

8.1.3 Audit

Audit Data Collection

The TOE provides the syslogd audit demon. This demon accepts log messages from a variety of other programs and writes them to the appropriate log files [AU.1].

Audit Events

Syslogd can record the following audit events:

a) Start-up and shutdown of the audit functions;

b) All use of the authentication mechanism, including the origin of the attempt and the identity provided during successful attempts;

c) Modification of user security attributes (including new values);

d) Attempts to revoke security attributes associated with users;

e) Modifications to the group of users that are associated with a role. [AU.2].

For all audited events the TSF can record the date and time of the event, the id of the associated process, and the identity of the user where relevant [AU.3].

The TSF can be configured only by an administrator to include or exclude auditable events from the set of audited events based on:

a) event severity;

b) type of event [AU.4].

Viewing of Audit Logs

Audit logs are viewed using a standard text editor. Text processing tools are provided for audit analysis that allow searches and sorting based on user identity and date/time [AU.5]. Only an administrator has access to the audit logs [AU.6]

Maintenance of Audit Logs

The TOE can be configured to periodically close audit logs and open up new ones, storing or deleting the old logs [AU.7].

Audit Data Loss

The ext3 file system is used to maintain data integrity in the event of unclean system shutdown. This file system can be configured to allow trade-offs between data integrity and speed of logging. See http://www.redhat.com/support/wpapers/redhat/ext3/index.html#integrity for further details [AU.8].

Time

The TOE hardware platform includes a real-time clock that is accessed by functions provided by the TSF. Users can obtain time and date information via these functions, but only an administrator can modify the date and time [AU.9].

8.1.4 Object Reuse

When an object is allocated to a subject from the TSF's pool of unallocated resources, the TSF will ensure that the object contains no data for which the subject is not authorised [OR.1]. The kernel clears each memory page before it is allocated to a process [OR.2]. Processes inherit all attributes from the parent process.

The TSF will revoke all access rights held by a subject to an object before allowing reuse by any other subject [OR.3].

8.1.5 Process Separation

The TSF enforces separation of user processes, and of the kernel from those user processes [PS.1]. This is achieved through use of a privileged execution mode provided by the hardware in which to run the kernel. Each process runs in a separate address space allocated by the kernel, and the execution of each process in that address space is controlled by the kernel scheduler [PS.2].

8.1.6 TSF Invocation

The TSF ensures that its security protection mechanisms cannot be bypassed. System resources are managed by the TSF, and can only be accessed through defined TSF interfaces [TI.1].

All resources that are managed by the kernel can only be accessed when running in kernel mode [TI.2]. User processes can only access kernel functions by means of exceptions or interrupts, in which case the kernel ensures that these functions are only used in pre-defined ways [TI.3].

Where system calls are intended for use only by trusted processes the kernel verifies that the calling process has an effective User id of 0 [TI.4].

8.1.7 Self testing

[ST1] not used.

The administrator is able to run a utility to search for bad blocks on a disk partition [ST.2].

8.2 Assurance Measures

This section identifies the Configuration Management, Delivery/Operation, Development, Guidance Documents, Test, and Vulnerability Assessment measures applied to satisfy CC assurance requirements.

TABLE 2: Assurance measures

Assurance Measure	**Security Assurance Requirement Met**
Documentation for the Red Hat configuration management system shows how Red Hat identifies and labels configuration items.	ACM_CAP.2
Red Hat Enterprise Linux delivery procedures describe how the TOE is delivered via secure download from https://rhs.redhat.com, and by physical delivery on CD.	ADO_DEL.1
Instructions for installation are provided in the Red Hat Enterprise Linux 3 Installation Guide. This is supplemented by further guidance on achieving the evaluated configuration.	ADO_IGS.1
A functional specification is provided that describes all system calls, trusted commands and related configuration files. Much of the information is given by reference to man pages.	ADV_FSP.1
A high-level design is provided that describes the subsystems that provide the security functions of the product.	ADV_HLD.1

Correspondence information is provided that maps the security functions in the ST to the functional specification and the high-level design.	ADV_RCR.1
A set of reference manuals is provided with the product. These manuals are supported by comprehensive man files	AGD_ADM.1, AGD_USR.1
Test plans and procedures are provided for the TSF, documented to a level where tests can be repeated. Expected and actual test results are supplied. Hardware is provided to the evaluators to allow tests to be repeated and additional tests to be run.	ATE_COV.1, ATE_FUN.1, ATE_IND.2
A strength of function analysis is provided for the TOE authentication function.	AVA_SOF.1
A vulnerability analysis is provided that documents a search for vulnerabilities in the TOE. This search is based on available documentation and public domain sources.	AVA_VLA.1

The above table includes all of the assurance requirements for the target level of assurance EAL2. Documented evidence covering each of the detailed security assurance requirements in EAL2 will be provided in the supporting documentation listed above against each EAL2 component.

9.0 Rationale

This section provides the rationale for the selection, creation, and use of the security policies, objectives, and components. Section 9.1 provides the rationale for the existence of the security objectives based upon the stated security policies while Section 9.2 provides the lower-level rationale for the existence of functional and assurance components based upon the stated security objectives. Section 9.2 provides an analysis that maps given security objectives to components as well as mapping given components to security objectives. In providing a mapping in both directions for the components and objectives, assurance is gained that the objectives were entirely met. This is further detailed in Section 9.2.

In addition to providing a complete rationale, Section 5 also provides the necessary application notes needed to understand how a TOE must meet the stated security objectives. These application notes provide additional information about a particular family/component/element that a developer or evaluator may need in order to fully understand how the component is to be applied.

Section 9.3 provides a table to demonstrate that all dependencies between security functional components have been met.

Section 9.4 provides a rationale for the TOE summary specification, and sections 9.5 and 9.6 provide rationales for the assurance and SOF ratings, respectively.

9.1 Security Objectives Rationale

This section provides a rationale for the existence of each threat, policy statement, security objective, and component that comprise the protection profile.

9.1.1 Complete Coverage - Threats

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

9.1.2 Complete Coverage - Policy

This section provides evidence demonstrating coverage of the Organisational Security Policy by both the IT and Non-IT security objectives. The following table shows this objective to policy mapping, and the table is followed by a discussion of the coverage for each Security Policy.

TABLE 3: Mapping of OSPs to Security Objectives

Organisational Security Policy	**Security Objectives**
P.AUTHORISED_USERS	O.AUTHORIZATION O.MANAGE O.ENFORCEMENT
P.NEED_TO_KNOW	O.DISCRETIONARY_ACCESS O.RESIDUAL_INFORMATION O.MANAGE O.ENFORCEMENT
P.ACCOUNTABILITY	O.AUDITING O.MANAGE O.ENFORCEMENT

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

P.AUTHORISED_USERS
Only those users who have been authorised to access the information within the system may access the system.
This policy is implemented by the O.AUTHORIZATION objective. The O.MANAGE supports this policy by requiring authorised administrators to be able to manage the functions and O.ENFORCEMENT ensures that functions are invoked and operate correctly.

P.NEED_TO_KNOW
The system must limit the access to, modification of, and destruction of the information in protected resources to those authorised users which have a “need to know” for that information.
This policy is implemented by the O.DISCRETIONARY_ACCESS objective. The O.RESIDUAL_INFORMATION objective ensures that information will not given to users who do not have a need to know, when resources are reused. The O.MANAGE supports this policy by requiring authorised administrator be able to manage the functions and O.ENFORCEMENT ensures that functions are invoked and operate correctly.

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

9.1.3 Complete Coverage - Environmental Assumptions

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

TABLE 4: Mapping of Non-IT Security Objectives to Environmental Assumptions

Non IT Security Objectives	Environmental Assumptions
O.INSTALL	A.MANAGE A.NO_EVIL_ADM A.PEER This objective requires those responsible for the TOE to ensure that the TOE is managed and operated in a manner consistent with the IT objectives (A.MANAGE). This same objective places an obligation on those responsible to ensure that administrators are trustworthy and properly trained (A.NO_EVIL_ADM). The objective also requires the TOE to be installed and operated consistently with the guidance documentation, which refers to connectivity constraints (A.PEER)
O.PHYSICAL	A.LOCATE A.PROTECT A.CONNECT This objective addresses the need to locate the TOE in a physically secure environment

	(A.LOCATE). This also covers connections to other devices (A.CONNECT), and includes the requirement to protect hardware and software from unauthorised modification as a result of physical access (A.PROTECT).
O.CREDEN	A.COOP This objective seeks user co-operation in protecting user authentication credentials This is covered by the general objective that users co-operate to use the TOE in a secure manner (A.COOP).

From the above table it can be seen that the objectives are suitable to cover all of the identified assumptions.

9.2 Security Requirements Rationale

This section provides evidence supporting the combined internal consistency and completeness of the functional components that comprise the CAPP.

9.2.1 Internal Consistency of Requirements

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

The functional components were selected from pre-defined CC components. The use of component refinement was accomplished in accordance with CC guidelines. An additional component was included to clarify the relationship of objects and security attributes.

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

Multiple instantiation of identical or hierarchically-related components was used to clearly state the required functionality that must exist in the TOE.

9.2.2 Complete Coverage - Objectives

This section demonstrates that the functional components selected for this profile provide complete coverage of the defined security objectives. The mapping of components to security objectives is depicted in the following table.

TABLE 5: Correspondence of security objectives to Security Functional Requirements

Security Objective	**Functional Requirement**
O.AUTHORIZATION	5.3.1 User Attribute Definition (FIA_ATD.1) 5.3.2 Strength of Authentication Data (FIA_SOS.1) 5.3.3 Authentication (FIA_UAU.2) 5.3.4 Protected Authentication Feedback (FIA_UAU.7) 5.3.5 Identification (FIA_UID.2) 5.4.6 Management of Authentication Data (FMT_MTD.1)
O.DISCRETIONARY_ACCESS	5.2.1 Discretionary Access Control Policy (FDP_ACC.1) 5.2.2 Discretionary Access Control Functions (FDP_ACF.1) 5.3.1 User Attribute Definition (FIA_ATD.1)

	5.3.6 User-Subject Binding (FIA_USB.1) 5.4.1 Management of Object Security Attributes (FMT_MSA.1) 5.4.2 Static Attribute Initialisation (FMT_MSA.3) 5.4.8 Revocation of Object Attributes (FMT_REV.1)
O.AUDITING	5.1.1 Audit Data Generation (FAU_GEN.1) 5.1.2 User Identity Association (FAU_GEN.2) 5.1.3 Audit Review (FAU_SAR.1) 5.1.4 Restricted Audit Review (FAU_SAR.2) 5.1.5 Selectable Audit Review (FAU_SAR.3) 5.1.6 Guarantees of Audit Data Availability (FAU_STG.1) 5.3.6 User-Subject Binding (FIA_USB.1) 5.4.3 Management of the Audit Trail (FMT_MTD.1) 5.4.4 Management of Audited Events (FMT_MTD.1) 5.5.4 Reliable Time Stamps (FPT_STM.1)
O.RESIDUAL_INFORMATION	5.2.3 Object Residual Information Protection (FDP_RIP.2) 5.2.4 Subject Residual Information Protection (FDP_RIP.3)
O.MANAGE	5.1.3 Audit Review (FAU_SAR.1) 5.1.5 Selectable Audit Review (FAU_SAR.3) 5.4.3 Management of the Audit Trail (FMT_MTD.1) 5.4.4 Management of Audited Events (FMT_MTD.1) 5.4.5 Management of User Attributes (FMT_MTD.1) 5.4.6 Management of Authentication Data (FMT_MTD.1) 5.4.7 Revocation of User Attributes (FMT_REV.1) 5.4.9 Security Management Functions (FMT_SMF.1) 5.4.10 Security Management Roles (FMT_SMR.1)
O.ENFORCEMENT	5.5.1 Abstract Machine Testing (FPT_AMT.1) 5.5.2 Reference Mediation (FPT_RVM.1) 5.5.3 Domain Separation (FPT_SEP.1)

The following discussion provides detailed evidence of coverage for each security objective:

O.AUTHORIZATION
The TSF must ensure that only authorised users gain access to the TOE and its resources.
Users authorised to access the TOE are defined using an identification and authentication process [5.3.5, 5.3.3]. To ensure authorised access to the TOE, authentication data is protected [5.3.1, 5.3.4, 5.4.6]. The strength of the authentication mechanism must be sufficient to ensure unauthorised users cannot easily pose as authorised users [5.3.2].

O.DISCRETIONARY_ACCESS
The TSF must control accessed to resources based on identity of users. The TSF must allow authorised users to specify which resources may be accessed by which users.
Discretionary access control must have a defined scope of control [5.2.1]. The rules of the DAC policy must be defined [5.2.2]. The security attributes of objects used to enforce the DAC policy must be defined. The security attributes of subjects used to enforce the DAC policy must be defined [5.3.1, 5.3.6]. Authorised users must be able to control who has access to objects [5.4.1] and be able to revoke that access [5.4.8]. Protection of named objects must be continuous, starting from object creation [5.4.2].

O.AUDITING
The TSF must record specified security relevant actions of users of the TOE. The TSF must present this information to authorised administrators.
Security-relevant actions must be defined, auditable [5.1.1], and capable of being associated with individual users [5.1.2, 5.3.6]. The audit trail must be protected so that

only authorised users may access it [5.1.4]. The TSF must provide the capability to audit the actions of an individual user [5.1.5, 5.3.6]. The audit trail must be complete [5.1.6]. The time stamp associated must be reliable [5.5.4]. An authorised administrator must be able to review [5.1.3] and manage [5.4.3, 5.4.4] the audit trail.

O.RESIDUAL_INFORMATION
The TSF must ensure that any information contained in a protected resource is not released when the resource is recycled.
Residual information associated with defined subjects and objects in the TOE must be purged prior to the reuse of the subject and object containing the residual information [5.2.3, 5.2.4].

O.MANAGE
The TSF must provide all the functions and facilities necessary to support the authorised
administrators that are responsible for the management of TOE security.
The TSF must provide for an authorised administrator to manage the TOE [5.4.10]. The administrator must be able to administer user accounts [5.4.5, 5.4.6, 5.4.7, 5.4.9]. The administrator must be able to review and manage the audit trail [5.1.3, 5.1.5, 5.4.3, 5.4.4, 5.4.9].

O.ENFORCEMENT
The TSF must be designed and implemented in a manner that ensures that the organisational policies are enforced in the target environment.
The TSF must make and enforce the decisions of the TSP [5.5.2]. It must be protected from interference that would prevent it from performing its functions [5.5.3]. Additionally, the TOE must provide the capability to demonstrate correct operation of the TSF's underlying abstract machine [5.5.1]. The correctness of this objective is further met through the assurance requirements defined in this PP.
This objective provides global support to other security objectives for the TOE by protecting the parts of the TOE which implement policies and ensures that policies are enforced.

9.3 Dependencies

The following table shows the dependencies that exist between functional components. A box with an X in it indicates a dependency that has been satisfied in the ST. A box with an O in it indicates an optional dependency where one of the options has been satisfied.

TABLE 6: Satisfaction of Dependencies

Section	CC Identifier	FAU_GEN.1	FAU_SAR.1	FAU_STG.1	FDP_ACC.1	FDP_ACF.1	FIA_ATD.1	FIA_UAU.2	FIA_UID.2	FMT_MSA.1	FMT_MSA.3	FMT_MTD.1	FMT_SMR.1	FMT_SMF.1	FPT_STM.1
5.1.1	FAU_GEN.1														X
5.1.2	FAU_GEN.2	X							X						
5.1.3	FAU_SAR.1	X													
5.1.4	FAU_SAR.2		X												
5.1.5	FAU_SAR.3		X												
5.1.6	FAU_STG.1	X													
5.2.1	FDP_ACC.1					X									
5.2.2	FDP_ACF.1				X						X				
5.2.3	FDP_RIP.2														
5.2.4	FDP_RIP.3														
5.3.1	FIA_ATD.1														
5.3.2	FIA_SOS.1														
5.3.3	FIA_UAU.2								X						
5.3.4	FIA_UAU.7							X							
5.3.5	FIA_UID.2														
5.3.6	FIA_USB.1						X								
5.4.1	FMT_MSA.1				O								X	X	
5.4.2	FMT_MSA.3									X			X		
5.4.3	FMT_MTD.1												X	X	
5.4.4	FMT_MTD.1												X	X	
5.4.5	FMT_MTD.1												X	X	
5.4.6	FMT_MTD.1												X	X	
5.4.7	FMT_REV.1												X		
5.4.8	FMT_REV.1												X		
5.4.9	FMT_SMF.1														
5.4.10	FMT_SMR.1								X						
5.5.1	FPT_AMT.1														
5.5.2	FPT_RVM.1														
5.5.3	FPT_SEP.1														
5.5.4	FPT_STM.1														

Note that dependencies on FIA_UID.1 are met by the inclusion of FIA_UID.2, since FIA_UID.2 is hierarchical to FIA_UID.1. Similarly, dependencies on FIA_UAU.1 are met by the inclusion of FIA_UAU.2, since FIA_UAU.2 is hierarchical to FIA_UAU.1.

9.4 TOE Summary Specification Rationale

This section demonstrates that the TOE security functions and assurance measures are suitable to meet the TOE security requirements.

The specified TOE security functions work together to satisfy the TOE security functional requirements. The following table demonstrates that each SFR is covered by at least one security function.

TABLE 7: Security functions address security requirements

Security Functional Requirement		Security Function
FAU_GEN.1	Audit data generation	AU.1, AU.2, AU.3, AU.9
FAU_GEN.2	Use identity association	AU.3
FAU_SAR.1	Audit review	AU.5
FAU_SAR.2	Restricted audit review	DAC.1
FAU_SAR.3	Selectable audit review	AU.4, AU.5
FAU_STG.1	Protected audit trail storage	DAC.1, AU.7, AU.8
FDP_ACC.1	Discretionary access control policy	DAC.1
FDP_ACF.1	Discretionary access control functions	DAC.3, DAC.4, DAC.5, DAC.8, TI.4
FDP_RIP.2	Object residual information protection	OR.1, OR.3
FDP_RIP.3	Subject residual information protection	OR.2
FIA_ATD.1	User attribute definition	IA.1, IA.2
FIA_SOS.1	Specification of secrets	IA.4, IA.5, IA.6, IA.7, IA.15
FIA_UAU.2	User authentication before any action	IA.4, IA.5, IA.8, IA.9, IA.12
FIA_UAU.7	Protected authentication feedback	IA.14
FIA_UID.2	User identification before any action	IA.4, IA.5
FIA_USB.1	User-subject binding	AU.3, DAC.2, IA.1, IA.10, IA.11
FMT_MSA.1	Management of security attributes	DAC.3, DAC.8
FMT_MSA.3	Static attribute initialisation	DAC.6, DAC.7
FMT_MTD.1(5.4.3)	Management of TSF data (1)	AU.6
FMT_MTD.1(5.4.4)	Management of TSF data (2)	AU.2, AU.4, AU.6
FMT_MTD.1(5.4.5)	Management of TSF data (3)	IA.3
FMT_MTD.1(5.4.6)	Management of TSF data (4)	IA.12, IA.13
FMT_REV.1 (U)	Revocation of user attributes	DAC.1, DAC.8, IA.16
FMT_REV.1 (O)	Revocation of object attributes	DAC.1. DAC.3, DAC.8
FMT_SMF.1	Security management functions	AU.2, AU.4, AU.7, DAC.8, IA.3, IA.7, IA.12, IA.13
FMT_SMR.1	Security roles	IA.1, IA.2, IA.7, IA.10,

		IA.11
FPT_AMT.1	Abstract machine testing	ST.2
FPT_RVM.1	Non-bypassability of the TSP	TI.1, TI.2, TI.3, TI.4
FPT_SEP.1	TSF domain separation	PS.1, PS.2
FPT_STM.1	Reliable time stamps	AU.9

The following table shows the security functional requirements that each security function addresses. The table shows that each security function is required to address at least one security functional requirement.

TABLE 8: SFRs contributed to by each security function

Security Function	Security Functional Requirements	Notes
IA.1	FIA_ATD.1, FIA_USB.1, FMT_SMR.1	Provides support for the requirement to maintain security attributes (FIA_ATD.1), to bind users to the subjects acting on their behalf by means of user Ids (FIA_USB.1), and supports roles by means of the group ID (FMT_SMR.1).
IA.2	FIA_ATD.1, FMT_SMR.1	Provision of groups, and the means to associate them with users (FIA_ATD.1), supports the association of users with specific roles (FMT_SMR.1)
IA.3	FMT_MTD.1(para 5.4.5), FMT_SMF.1	FMT_SMF.1 requires the ability to modify user attributes and group memberships, and FMT_MTD.1 restricts this to administrators.
IA.4	FIA_SOS.1, FIA_UAU.2, FIA_UID.2	Authentication of user identity meets the requirements for identification (FIA_UID.2) and authentication (FIA_UAU.2) before other actions can be carried out. FIA_SOS.1 requires that mechanism to be sufficiently strong to prevent unauthorised access.
IA.5	FIA_SOS.1, FIA_UID.2, FIA_UAU.2	Provides more detail in support of IA.4.
IA.6	FIA_SOS.1	Provides an additional protection measure against repeated password guessing attacks.
IA.7	FIA_SOS.1, FMT_SMF.1, FMT_SMR.1	Administrators (FMT_SMR.1) can place restrictions on the authentication function (FMT_SMF.1) that support its minimum strength (FIA_SOS.1).
IA.8	FIA_UAU.2	Implementation detail for the authentication function.
IA.9	FIA_UAU.2	Implementation detail for the authentication function.
IA.10	FMT_SMR.1, FIA_USB.1	Supports the operation of roles (FMT_SMR.1), and the association of users with actions taken in those roles (FIA_USB.1).
IA.11	FMT_SMR.1, FIA_USB.1	Supports the operation of roles (specifically root)(FMT_SMR.1), and the association of

		users with actions taken in those roles (FIA_USB.1).
IA.12	FIA_UAU.2, FMT_MTD.1(para 5.4.6), FMT_SMF.1	Provides the ability for administrators to change passwords (FMT_MTD.1, FMT_SMF.1), and general support for the authentication function (FIA_UAU.2).
IA.13	FMT_MTD.1(para 5.4.6), FMT_SMF.1	Provides the function to initialise passwords (FMT_SMF.1) and restricts it to an administrator (FMT_MTD.1)
IA.14	FIA_UAU.7	Provides protected feedback for password entry.
IA.15	FIA_SOS.1	Supports the strength of the authentication token by use of message digest.
IA.16	FMT_REV.1(U)	Supports the revocation of access rights by terminating a user session.
DAC.1	FAU_SAR.2, FAU_STG.1, FDP_ACC.1, FMT_REV.1(O), FMT_REV.1(U),	Provides mediation of access in support of the TSF (FDP_ACC.1), and thus protection of the audit trail (FAU_SAR.2, FAU_STG.1), and allows access to be revoked (FMT_REV.1(O), FMT_REV.1(U)).
DAC.2	FIA_USB.1	Provides mechanisms to control association of users with processes.
DAC.3	FDP_ACF.1, FMT_MSA.1, FMT_REV.1(O)	Provides detail on the mechanism used to grant or deny access to objects within the TSC.
DAC.4	FDP_ACF.1	Specific detail on special flags in support of the TSP.
DAC.5	FDP_ACF.1	Specific detail on how access rights are represented within the TOE.
DAC.6	FMT_MSA.3	Administrator control of default access permissions.
DAC.7	FMT_MSA.3	Handling of default permissions on file creation.
DAC.8	FDP_ACF.1, FMT_MSA.1, FMT_REV.1(O), FMT_REV.1(U), FMT_SMF.1	Ownership and access permissions can be changed (FMT_SMF.1) only by an administrator (FMT_MSA.1) or revoked (FMT_REV.1(O), FMT_REV.1(U)).
AU.1	FAU_GEN.1	Mechanism for recording audit information.
AU.2	FAU_GEN.1, FMT_MTD.1(para 5.4.4), FMT_SMF.1	Ability to record audit events (FAU_GEN.1) and to control changes to the list of events recorded (FMT_MTD.1, FMT_SMF.1).
AU.3	FAU_GEN.1, FAU_GEN.2, FIA_USB.1	Recording of date and time for audit events (FAU_GEN.1), and association of users with audited events (FAU_GEN.2, FIA_USB.1).
AU.4	FMT_MTD.1(para 5.4.4), FMT_SMF.1, FAU_SAR.3	The ability to select what is audited is provided (FMT_SMF.1, FAU_SAR.3), and is restricted to an administrator (FMT_MTD.1)
AU.5	FAU_SAR.1, FAU_SAR.3	Provides ability to read audit data (FAU_SAR.1) and to process it (FAU_SAR.3).

AU.6	FMT_MTD.1(para 5.4.3)	Restricts access to the audit records to an administrator.
AU.7	FAU_STG.1, FMT_SMF.1	Supports the ability to manage storage of audit records.
AU.8	FAU_STG.1	Support for the integrity of audit records.
AU.9	FAU_GEN.1, FPT_STM.1	The TOE can generate date/time information (FPT_STM.1) for association with audited events (FAU_GEN.1).
OR.1	FDP_RIP.2	Removing access to data before objects are reused.
OR.2	FDP_RIP.3	Ensuring memory is cleared before being allocated to a process.
OR.3	FDP_RIP.2	Prevents further access by a subject to an object once it is in use by another subject.
PS1	FPT_SEP.1	Provides separation of the kernel from user processes.
PS.2	FPT_SEP.1	Provides the mechanism to support separation.
TI.1	FPT_RVM.1	Prevention of bypass by restricting access to defined interfaces.
TI.2	FPT_RVM.1	No user access to kernel resources is permitted.
TI.3	FPT_RVM.1	Provides mechanism to enforce prevention of bypass.
TI.4	FDP_ACF.1, FPT_RVM.1	Provides mechanism to enforce prevention of bypass.
ST.2	FPT_AMT.1	Function to test disk storage.

The rationale for the TOE security functional requirements given in Section 5 and elaborated in Section 9.2, demonstrates that the SFRs work together in a consistent manner and are mutually supportive. Given that the above tables show that the security functions are completely instantiate the SFRs, and that they contain no conflicting or inconsistent requirements, it is determined that the security functions do not introduce any security weaknesses.

9.5 Rationale for Assurance Rating

This security target has been developed for a generalised environment with a moderate level of risk to the assets. It is intended that products used in these environments will be generally available, without modification to meet the security needs of the environment. As such it was determined that Evaluation Assurance Level 2 was the most appropriate.

9.6 Rationale for SOF Rating

The strength of function rating of SOF-medium is consistent with the SFR FIA_SOS.1 by providing a ‘one off’ probability of guessing the password of less than 1 in 1,000,000. This SFR is in turn consistent with the security objectives described in Section 4.

10.0 Notes on Deviations from CC

This section contains notes on places where this security target deviated from version 2.1 of the Common Criteria. These deviations follow those used in CAPP.

Note 1 The CC's FDP_RIP components only specify resources being allocated to objects and do not address resources used directly by subjects, such as memory or registers. The explicit requirement FDP_RIP.3 was added to ensure coverage of these resources. The words are identical to FDP_RIP.2 except "subject" replaces "object".

Note 2 The CC's FIA_USB component used the term "appropriate security attributes", which is really too vague. The word "appropriate" was replaced with the word "following" and an assignment list was added. This allows the ST to specify what attributes are needed to enforce the TSP. In addition, elements were added to cover rules that are required to be enforced on attribute binding or changes.

Note 3 The format of using sub-elements which appeared in the CC's FAU_GEN.1 was difficult to represent all the information in a clear fashion. The sub-elements were replaced by the use of a table, as the wording of the element adjusted to refer to the table, rather than the sub-elements.