Red Hat Enterprise Linux 5 Security Target for CAPP, LSPP and RBAC compliance

Version: 3.9

Last Update: 2007-05-31

Document History

Version	Date	Changes	Summary	Author
2.0	2005-09-16	n/a	Initial version for RHEL4 U2 and NIAP scheme	Klaus Weidner, atsec
2.1	2005-10-12	minor	minor updates	Klaus Weidner, atsec
2.2	2005-12-12	minor	updated package list; minor updates based on evaluator's feedback	Klaus Weidner, atsec
2.3	2005-12-14	minor	Updated audit events table	Klaus Weidner, atsec
3.0	2006-03-09	major	TOE is RHEL 5; Added LSPP and RBAC modes	Gerald Krummeck, atsec
3.1	2006-03-22	major	Update of Rationale, minor errors	Gerald Krummeck, atsec
3.2	2006-04-20	minor	Addressing comments from CB	Gerald Krummeck, atsec
3.3	2006-05-31	minor	Updated with hardware changes	Robert Wenner, atsec
3.4	2006-08-28	minor	Removed at package	Robert Wenner, atsec
3.5	2006-09-26	minor	Various updates	Robert Wenner, atsec
3.6	2006-10-18	minor	Updates due to HLD	Wolfgang Mauerer, atsec
3.7	2007-02-06	minor	Updates due to LLD, adding IPv6	Wolfgang Mauerer, atsec
3.8	2007-05-08	minor	Updates after finishing of development of TOE	Wolfgang Mauerer, atsec
3.9	2007-05-31	minor	Add Client to platform list	Wolfgang Mauerer, atsec

Table of Content

1 Introduction....8
1.1 ST Identification....8
1.2 ST Overview....8
1.3 CC Conformance....9
1.4 Strength of Function....9
1.5 Structure....9
1.6 Terminology....9
2 TOE Description....11
2.1 Intended Method of Use....11
2.2 Summary of Security Features....12
2.2.1 Identification and Authentication....13
2.2.2 Audit....13
2.2.3 Discretionary Access Control....13
2.2.4 Mandatory Access Control (LSPP/RBAC mode only)....13
2.2.5 Role-based Access Control (LSPP/RBAC mode only)....13
2.2.6 Object Reuse....14
2.2.7 Security Management....14
2.2.8 Secure Communication....14
2.2.9 TSF Protection....14
2.3 Software....14
2.4 Configurations....21
2.4.1 File systems....22
2.4.2 TOE hardware....22
3 TOE Security Environment....24
3.1 Introduction....24
3.2 Threats....24
3.2.1 Threats countered by the TOE....24
3.2.2 Threats to be countered by measures within the TOE environment....25
3.3 Organizational Security Policies....25
3.4 Assumptions....26
3.4.1 Physical Aspects....26
3.4.2 Personnel Aspects....26
3.4.3 Procedural Assumptions....26
3.4.4 Connectivity Aspects....26
4 Security Objectives....28
4.1 Security Objectives for the TOE....28
4.2 Security Objectives for the TOE Environment....28

5 Security Requirements 30
5.1 TOE Security Functional Requirements 30
5.1.1 Security Audit (FAU) 30
5.1.2 Cryptographic Support (FCS) 37
5.1.3 User Data Protection (FDP) 39
5.1.4 Identification and Authentication (FIA) 46
5.1.5 Security Management (FMT) 48
5.1.6 Protection of the TOE Security Functions (FPT) 52
5.1.7 TOE Access (FTA) 54
5.1.8 Trusted Path/Channels (FTP) 54
5.1.9 Strength of Function 54
5.2 TOE Security Assurance Requirements 54
5.3 Security Requirements for the IT Environment 55
5.4 Security Requirements for the Non-IT Environment 55
6 TOE Summary Specification 56
6.1 Security Enforcing Components Overview 56
6.1.1 Introduction 56
6.1.2 SELinux 56
6.1.3 Kernel Services 58
6.1.4 Non-Kernel TSF Services 59
6.1.5 Network Services 60
6.1.6 Security Policy Overview 60
6.1.7 TSF Structure 61
6.1.8 TSF Interfaces 61
6.1.9 Secure and Non-Secure States 62
6.2 Description of the Security Enforcing Functions 63
6.2.1 Introduction 63
6.2.2 Identification and Authentication (IA) 63
6.2.3 Audit (AU) 66
6.2.4 Discretionary Access Control (DA) 68
6.2.5 Mandatory Access Control (MA) (LSPP/RBAC mode only) 74
6.2.6 Role-based Access Control (RBAC) (LSPP/RBAC mode only) 76
6.2.7 Object Reuse (OR) 77
6.2.8 Security Management (SM) 78
6.2.9 Secure Communication (SC) 81
6.2.10 TSF Protection (TP) 84
6.3 Supporting functions part of the TSF 89
6.3.1 Processes executed by non-administrative users 89
6.4 Assurance Measures 90

6.5 TOE Security Functions requiring a Strength of Function 91
7 Protection Profile Claims 92
7.1 PP Reference 92
7.2 PP Tailoring 92
7.2.1 Security Functional Requirements 92
7.2.2 Threats, Policies, Assumptions and Objectives 92
7.2.3 Assurance Requirements 93
8 Rationale 94
8.1 Security Objectives Rationale 94
8.1.1 Security Objectives Coverage 94
8.1.2 Security Objectives Sufficiency 95
8.2 Security Requirements Rationale 97
8.2.1 Internal Consistency of Requirements 97
8.2.2 Security Requirements Coverage 103
8.2.3 Security Requirements Dependency Analysis 106
8.2.4 Strength of function 108
8.2.5 Evaluation Assurance Level 108
8.3 TOE Summary Specification Rationale 109
8.3.1 Security Functions Justification 109
8.3.2 Assurance Measures Justification 113
8.3.3 Strength of function 113
8.3.4 PP Threats 114
8.3.5 PP Assumptions 115
8.3.6 PP Objectives 116
8.3.7 PP SFRs 117
9 Abbreviations 121

References

[CAPP]	Controlled Access Protection Profile, Issue 1.d, 8 October 1999
[CC]	Common Criteria for Information Technology Security Evaluation, Parts 1 to 3, CCMB-2005-08-001 to CCMB-2005-08-003, Version 2.3, August 2005
[CEM]	Common Methodology for Information Technology Security Evaluation, Evaluation Methodology, CCMB-2005-08-004, Version 2.3, August 2005
[ECG]	Evaluated Configuration Guide
[GUIDE]	ISO/IEC PDTR 15446 Title: Information technology – Security techniques – Guide for the production of protection profiles and security targets, ISO/IEC JTC 1/SC 27 N 2449, 2000-01-04
[IPSEC]	“Security Architecture for the Internet Protocol”, ftp://ftp.rfc-editor.org/in-notes/rfc2401.txt
[LSPP]	Labeled Security Protection Profile, Version 1.b, 8 October 1999
[RBACPP]	Role-Based Access Control Protection Profile, version 1.0, dated July 30, 1998
[SSH-AUTH]	RFC 4252: The Secure Shell (SSH) Authentication Protocol, http://www.ietf.org/rfc/rfc4252.txt
[SSH-TRANS]	Ylonen, T. and C. Lonvick, Ed., "The Secure Shell (SSH) Transport Layer Protocol", RFC 4253, January 2006.[SSLv3] The SSL Protocol Version 3.0; http://wp.netscape.com/eng/ssl3/drft302.txt
[TARGET]	Red Hat Enterprise Linux 5 Security Target (this document)
[TLS-AES]	RFC 3268: Advanced Encryption Standard (AES) Ciphersuites for Transport Layer Security (TLS), http://www.ietf.org/rfc/rfc3268.txt
[X.509]	ITU-T RECOMMENDATION X.509 | ISO/IEC 9594-8: INFORMATION TECHNOLOGY - OPEN SYSTEMS INTERCONNECTION - THE DIRECTORY: PUBLIC-KEY AND ATTRIBUTE CERTIFICATE FRAMEWORKS
[RFC2104]	H. Krawczyk, M. Bellare, R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997

1 Introduction

This is version 3.9 of the Security Target document for the evaluation of Red Hat Enterprise Linux Version 5 Server and Client (RHEL). There are no technical differences between the "Server" and "Client" version.

RHEL in the evaluated configuration does not contain an X11 server and therefore no X11-based applications. The product is configured to be used as a server.

The TOE includes the hardware and firmware used to run the software components.

1.1 ST Identification

Title: Red Hat Enterprise Linux 5 Security Target for CAPP, LSPP and RBAC compliance, Version 3.9

Keywords: Linux, Open Source, general-purpose operating system, POSIX, UNIX, security, multi-level security, role-based access control.

This document is the security target for the CC evaluation of the Red Hat Enterprise Linux 5 operating system product, and is conformant to the Common Criteria for Information Technology Security Evaluation [CC] with extensions as defined in the Controlled Access Protection Profile [CAPP] and the Labeled Security Protection Profile [LSPP].

1.2 ST Overview

This security target documents the security characteristics of the Red Hat Enterprise Linux 5 operating system.

Red Hat Enterprise Linux is a highly-configurable Linux-based operating system which has been developed to provide a good level of security as required in commercial environments and as a basis for secure computing, including

- multi-level security based on sensitivity labels for objects and clearances for subjects, implementing the Bell/LaPadula model of mandatory access control
- least-privilege operations using a fine-grained access control model and a rights and privilege management based on roles,
- secure communications over public communication channels using encrypted tunnels.

It also meets all of the requirements of

- the Controlled Access Protection Profile [CAPP] developed by the Information Systems Security Organization within the National Security Agency to map the TCSEC C2 class of the U.S. Department of Defence (DoD) Trusted Computer System Evaluation Criteria (TCSEC) to the Common Criteria framework
- the Labeled Security Protection Profile [LSPP] developed by the Information Systems Security Organization within the National Security Agency to map the TCSEC B1 class of the U.S. Department of Defence (DoD) Trusted Computer System Evaluation Criteria (TCSEC) to the Common Criteria framework
- the Role-Based Access Control Protection Profile [RBACPP] developed by NIST and CygnaCom Solutions.

This Security Target therefore claims full compliance with the requirements of these Protection Profiles and also includes additional functional and assurance packages beyond those required by CAPP, LSPP and RBAC.

Several servers running Red Hat Enterprise Linux can be connected to form a networked system. The communication aspects within Red Hat Enterprise Linux used for this connection are also part of the evaluation. Communication links can be protected against loss of confidentiality and integrity by security functions of the TOE based on cryptographic protection mechanisms.

This evaluation focuses on the use of the TOE as a server or a network of servers. Therefore a graphical user interface has not been included as part of the evaluation. In addition the evaluation assumes the operation of the network of servers in a non-hostile environment.

This Security Target covers two modes of operation of the TOE: In LSPP/RBAC mode, the TOE operates with all multi-level security and RBAC features enabled, thus fulfilling the requirements of [LSPP], [CAPP], and [RBACPP]. In CAPP mode, the TOE operates in a "standard" mode without these security extensions enabled, still meeting the requirements of [CAPP]. In CAPP mode, SELinux is either turned off or can be used with the targeted policy as

SELinux only has additional restrictions compared to CAPP. However, the MLS policy should not be used as it interferes with the UID 0 mechanics that are important to CAPP.

1.3 CC Conformance

This ST is CC *Part 2 extended* and *Part 3 conformant*, with a claimed Evaluation Assurance Level of EAL4 augmented by ALC_FLR.3.

The extensions to part 2 of the Common Criteria are those introduced by the Controlled Access Protection Profile [CAPP], the Labeled Security Protection Profile [LSPP] and the Role-Based Access Control Protection Profile [RBACPP].

1.4 Strength of Function

The claimed strength of function for this TOE is: SOF-medium.

1.5 Structure

The structure of this document is as defined by [CC] Part 1 Annex C.

- Section 2 is the TOE Description.
- Section 3 provides the statement of TOE security environment.
- Section 4 provides the statement of security objectives.
- Section 5 provides the statement of IT security requirements.
- Section 6 provides the TOE summary specification, which includes the detailed specification of the IT Security Functions.
- Section 7 provides the Protection Profile claim
- Section 8 provides the rationale for the security objectives, security requirements and the TOE summary specification.

1.6 Terminology

This section contains definitions of technical terms that are used with a meaning specific to this document. Terms defined in the [CC] are not reiterated here, unless stated otherwise.

Authorized user: This term refers to a user who has been properly identified and authenticated. These users are considered to be legitimate users of the TOE.

Administrative User: This term refers to an administrator of a RHEL. Administrators are users having privileges to execute special commands interfering with the security functionality or they can modify the configuration which affects the security functionality. In CAPP mode, the only administrative user is root (UID 0). In LSPP/RBAC mode, the role functionality allows different users having different administrative privileges.

Authorized administrator: An authorized administrator is an authorized 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. For the purpose of this ST, “authorized administrators” and “administrative users” are synonym terms.

Active Role Set (ARS): This is the subset of the set of authorized roles for a user that has actually been activated for the user in a particular user session. The total set of access rights (privileges) available to a user in a session is the sum of the access rights directly assigned to each member of ARS together with the privileges inherited by each member of ARS through roles assigned to it.

Authentication data: This includes the password for each user of the product. Authentication mechanisms using other authentication data than a password are not supported in the evaluated configuration.

Authorized Roles for the User: This is the set of roles directly assigned to the user by the RBAC Administrator together with the set of roles contained in those roles (due to role to role assignments)

Data: arbitrary bit sequences in computer memory or on storage media.

Default Active Role Set (DARS): Instead of forcing the user to build an Active Role Set (ARS) during every user session, the RBAC administrator provides a default set of roles (from the list of authorized roles for the user). The composition of DARS determines the initial available access rights for the user at the start of the session. In other words, DARS is the ARS at the time of session creation. In many software environment the user may be able to change the composition of this initial ARS (i.e. DARS) during the course of the user session.

Information: any data held within a server, including data in transit between servers.

Named Object: In Red Hat Enterprise Linux, those objects that are subject to access control, which are file system objects and IPC objects.

Object Owner: The user who creates a *named object* becomes the object owner by default. In some environments, the object owner can be changed by the system administrator. The object owner has generally all discretionary access rights on the object and the power to grant discretionary access rights on the objects he/she owns to roles and other users.

Object: In Red Hat Enterprise Linux, objects belong to one of three categories: file system objects, IPC objects, and memory objects.

Privilege Set for a Role: The total set of system privileges and access rights on various objects granted to a role.

Product: The term product is used to define software components that comprise the Red Hat Enterprise Linux system.

RHEL: This term serves as an abbreviation for "Red Hat Enterprise Linux", which is the Target of this evaluation.

Role: A role represents a set of actions that an authorized user, upon assuming the role, can perform. In this TOE only the roles of administrative user and normal user are supported.

Security Attributes: As defined by functional requirement FIA_ATD.1, the term ‘security attributes’ includes the following as a minimum: user identifier; group memberships; user authentication data.

SELinux: SELinux is a component of the Linux operating system implementing the MLS and role based access control checks. It is implemented in kernel space with supporting user space configuration tools. The kernel component utilizes the Linux Security Module framework to allow its functionality being disabled without impacting the rest of the kernel (SELinux is optional in CAPP mode).

Subject: There are two classes of subjects in Red Hat Enterprise Linux:

- untrusted internal subject - this is a Red Hat Enterprise Linux process running on behalf of some user, running outside of the TSF (for example, with no privileges).
- trusted internal subject - this is a Red Hat Enterprise Linux process running as part of the TSF. Examples are service daemons and the process implementing the identification and authentication of users.

System: Includes the hardware, software and firmware components of the Red Hat Enterprise Linux product which are connected/networked together and configured to form a usable system.

Target of Evaluation (TOE): The TOE is defined as the Red Hat Enterprise Linux operating system, running and tested on the hardware and firmware specified in this Security Target. The BootPROM firmware as well as the hardware form part of the TOE as required by the NIAP interpretation of CAPP.

User: Any individual/person who has a unique user identifier and who interacts with the Red Hat Enterprise Linux product.

2 TOE Description

The target of evaluation (TOE) is the operating system Red Hat Enterprise Linux 5.

Red Hat Enterprise Linux is a general purpose, multi-user, multi-tasking Linux based operating system. It provides a platform for a variety of applications in the governmental and commercial environment. Red Hat Enterprise Linux is available on a broad range of computer systems, ranging from departmental servers to multi-processor enterprise servers.

The Red Hat Enterprise Linux evaluation covers a potentially distributed, but closed network of HP (Itanium2, Pentium, Xeon, and Opteron based) servers running the evaluated version of Red Hat Enterprise Linux. The hardware platforms selected for the evaluation consist of machines which are available when the evaluation has completed and to remain available for a substantial period of time afterwards.

The TOE Security Functions (TSF) consist of functions of Red Hat Enterprise Linux that run in kernel mode plus some trusted processes. These are the functions that enforce the security policy as defined in this Security Target. Tools and commands executed in user mode that are used by an administrative user need also to be trusted to manage the system in a secure way. But as with other operating system evaluations they are not considered to be part of this TSF.

The hardware and the BootProm firmware are considered to be part of the TOE as required by the deliberate NIAP interpretation of CAPP and LSPP.

The TOE includes installation from CDROM/DVDROM and from a local hard disk partition.

The TOE includes standard networking applications, such as ftp, ssl and ssh. xinetd is used to protect network applications which might otherwise have security exposures. The TOE provides means to encrypt communication channels. IPSec allows transporting sensitivity labels, thus enabling to enforce mandatory access controls between connected systems providing the same implementation.

System administration tools include the standard Linux commands. A graphical user interface for system administration or any other operation is not included in the evaluated configuration.

The TOE environment also includes applications that are not evaluated, but are used as unprivileged tools to access public system services. For example a HTTP server using a port above 1024 (e. g. on port 8080) may be used as a normal application running without root privileges on top of the TOE. The Evaluated Configuration Guide provides guidance how to set up an HTTP server on the TOE in a secure way.

In its evaluated configuration, the TOE allows two modes of operation: LSPP/RBAC-compliant and CAPP-compliant. In both modes, the same software elements are used. While the CAPP-compliant mode is compliant to [CAPP] only, LSPP-compliant mode provides compliance to [LSPP], [RBACPP] and [CAPP].

2.1 Intended Method of Use

The TOE is a Linux-based multi-user multi-tasking operating system. The TOE may provide services to several users at the same time. After successful login, the users have access to a general computing environment, allowing the start-up of user applications, issuing user commands at shell level, creating and accessing files. The TOE provides adequate mechanisms to separate the users and protect their data. Privileged commands are restricted to administrative users.

The TOE uses a role-based model of normal (unprivileged) users and administrative users that have the capability to use certain privileges depending on their assigned role. The granularity of privilege assignments to administrative users varies between the modes of operation:

- In CAPP mode, the system allocates all privileges to the user ID 0 (initially allocated to the “root” account). A user allowed to switch to this identity (and thereby become an administrative user) can therefore exercise all these privileges. In addition, certain privileges (such as setting access rights for a file) are also available to the object owner.
- In LSPP/RBAC mode, privileges are assigned to certain roles. Users allowed to assume such a role are restricted to the privileges allocated to this role. Therefore, the power of the superuser can be broken up and assigned to different users, thus avoiding the concentration of all power in the hand of one administrator.

The TOE is intended to operate in a networked environment with other instantiations of the TOE as well as other well-behaved client systems operating within the same management domain. All those systems need to be configured in accordance with a defined common security policy.

The TOE permits one or more processors and attached peripheral and storage devices to be used by multiple users to perform a variety of functions requiring controlled shared access to the data stored on the system. Such installations are typical for workgroup or enterprise computing systems accessed by users local to, or with otherwise protected access to, the computer system.

It is assumed that responsibility for the safeguarding of the data protected by the TOE can be delegated to the TOE users for the purpose of discretionary access controls to user-owned data. All data is under the control of the TOE. The data is stored in named objects, and the TOE can associate with each named object a description of the access rights to that object.

All individual users are assigned a unique user identifier within the single host system that forms the TOE. This user identifier is used as the basis for discretionary access control decisions. The TOE authenticates the claimed identity of the user before allowing the user to perform any further actions.

The TOE enforces controls such that access to data objects can only take place in accordance with the access restrictions placed on that object by its owner or administrative users. Ownership of named objects may be transferred under the control of the access control policy.

Access rights (e.g. read, write, execute) can be assigned to data objects with respect to subjects (users). Once a subject is granted access to an object, the content of that object may be freely used to influence other objects accessible to this subject.

Red Hat Enterprise Linux has significant security extensions compared to standard UNIX systems:

- SELinux and LSM, providing a framework for domain-type access control, with role-based access control
- Access Control Lists for fine-grained access controls to persistent objects, allowing controls beyond the capabiligties of the traditional UNIX access control mechanism based on permission bits (to which the implementation is compatible)
- A Journaling File System
- Integrated authentication framework (PAM). The combination of PAM modules described in section 6.2.2 allows to enforce password quality metrics, to restrict logins from certain accounts by their point of entry, and to block logins from accounts after a number of consecutive failed logins.
- A dedicated auditing subsystem. This auditing subsystem allows for the auditing of security critical events and provides tools for the administrative user to configure the audit subsystem and evaluate the audit records.
- Basic check functions for the TOE's underlying abstract machine. They allow an administrative user to check on demand if the basic security functions of the hardware the TOE relies upon are provided correctly.

2.2 Summary of Security Features

The primary security features of the product are:

- Identification and Authentication
- Audit
- Discretionary Access Control
- Mandatory Access Control (LSPP/RBAC mode only)
- Role-based Access Control (LSPP/RBAC mode only)
- Object reuse functionality
- Security Management
- Secure Communication
- TSF Protection.

These primary security features are supported by domain separation and reference mediation, which ensure that the features are always invoked and cannot be bypassed.

2.2.1 Identification and Authentication

Red Hat Enterprise Linux provides identification and authentication using pluggable authentication modules (PAM) based upon user passwords. The quality of the passwords used can be enforced through configuration options controlled by Red Hat Enterprise Linux. Other authentication methods (e. g. Kerberos authentication, token based authentication) that are supported by Red Hat Enterprise Linux as pluggable authentication modules are not part of the evaluated configuration. Functions to ensure medium password strength and limit the use of the su command and restrict administrator login to specific terminals are also included.

2.2.2 Audit

The TOE provides an audit capability that allows generating audit records for security critical events. The administrative user can select which events are audited and for which users auditing is active. A list of events that can be audited is defined in chapter 5 and 6.

The TOE provides tools that help the administrative user extract specific types of audit events, audit events for specific users, audit events related to specific file system objects or audit events within a specific time frame from the overall audit records collected by the TOE. The audit records are stored in ASCII text, no conversion of the information into human readable form is necessary.

The audit function detects when the capacity of the audit trail exceeds configurable thresholds, and the system administrator can define actions to be taken when the threshold is exceeded. The possible actions include generating a syslog message to inform the administrator, switching the system to single user mode (this prevents all user-initiated auditable actions), or halting the system.

The audit function also ensures that no audit records get lost due to exhaustion of the internal audit buffers. Processes that try to create an audit record while the internal audit buffers are full will be halted until the required resources are available again. In the unlikely case of unrecoverable resource exhaustion, the kernel audit component initiates a kernel panic to prevent all further auditable events.

2.2.3 Discretionary Access Control

Discretionary Access Control (DAC) restricts access to file system objects based on Access Control Lists (ACLs) that include the standard UNIX permissions for user, group and others. Access control mechanisms also protect IPC objects from unauthorized access.

Red Hat Enterprise Linux includes the ext3 file system, which supports POSIX ACLs. This allows defining access rights to files within this type of file system down to the granularity of a single user.

2.2.4 Mandatory Access Control (LSPP/RBAC mode only)

Red Hat Enterprise Linux provides mandatory access control (MAC) in LSPP mode, which imposes access restrictions to information based on security classification. Users and resources can have a sensitivity label associated. Sensitivity labels contain a hierarchical classification (security level), which specify the sensitivity (for example: public, internal use, or secret), and zero or more non-hierarchical security categories.

The access control enforced by the TOE ensures that users can only read labeled information if their sensitivity labels dominate the information's label, and that they can only write to labeled information containers if the container's label dominates the subject's, thus implementing the Bell-LaPadula model of information flow control.

2.2.5 Role-based Access Control (LSPP/RBAC mode only)

Red Hat Enterprise Linux supports the concept of Roles, allowing administrative powers to be broken into many discrete Roles. This removes the requirement of one superuser (root or only one system-administrator) to administer the TOE and introduces a separation of duty. A Role combines a set of privileges to accomplish distinguished

administrative tasks, thus allowing the administrative functionality to be distributed and hence diluted amongst the Roles, to reduce the impact of any misuse of a Role.

Roles are also used in combination with the domain/type enforcement do define policies against such roles rather than for each individual separately.

2.2.6 Object Reuse

File system objects as well as memory and IPC objects will be cleared before they can be reused by a process belonging to a different user.

2.2.7 Security Management

The management of the security critical parameters of the TOE is performed by administrative users. A set of commands that require privileges are used for system management; they require users to possess appropriate privileges to execute them. Security parameters are stored in specific files that are protected by the access control mechanisms of the TOE against unauthorized access by users that are not administrative users.

2.2.8 Secure Communication

The TOE supports secure communication with other systems via the SSH v2, SSL v3, CIPSO, and IPSec protocols. Communication via those protocols is protected against unauthorized disclosure and modification via cryptographic mechanisms. The TOE also allows for secure authentication of the communicating parties using the SSL v3 protocol with client and server authentication. This allows establishing a secure communication channel between different machines running the TOE even over an insecure network. The SSL v3, CIPSO, and IPSec protocols can be used to tunnel otherwise unprotected protocols in a way that allows an application to secure its TCP based communication with other servers (provided the protocol uses a single TCP port).

2.2.9 TSF Protection

While in operation, the kernel software and data are protected by the hardware memory protection mechanisms. The memory and process management components of the kernel ensure a user process cannot access kernel storage or storage belonging to other processes.

Non-kernel TSF software and data are protected by DAC, MAC and process isolation mechanisms. In the evaluated configuration, the reserved user ID root owns the directories and files that define the TSF configuration. In general, files and directories containing internal TSF data (e.g., configuration files, batch job queues) are also protected from reading by DAC and MAC permissions.

The TOE including the hardware and firmware components are required to be physically protected from unauthorized access. The TOE contains two types of hardware components: those directly accessible to user processes (a subset of the CPU registers and memory); and those that the kernel protects from direct access by user programs. A user process may execute unprivileged instructions and read or write to memory and processor registers within the bounds defined by the kernel for the user process, those types of access are not mediated by the kernel. All other types of access to hardware resources by user processes can only be performed by requests (in the form of system calls) to the kernel.

The TOE provides a tool that allows an administrative user to check the correct operation of the underlying hardware. This tool performs tests to check the system memory, the memory protection features of the underlying processor and the correct separation between user and supervisor state.

2.3 Software

The Target of Evaluation is based on the following system software:

Red Hat Enterprise Linux 5

The TOE and its documentation are supplied on CD-ROM and via the Red Hat Network Internet delivery method. With the TOE software, the user receives the Evaluated Configuration Guide and scripts that can be used for the secure installation process. The user needs to verify the integrity and authenticity of those packages using the standard package verification procedure as described in the manuals distributed with the product.

The following list of packages makes up the TOE in the evaluated configuration. This includes packages that contribute to the TSF as well as packages that contain untrusted user programs from the distribution. Note that additional untrusted user programs may be installed and used as long as they are

- not SUID or SGID to root;
- (LSPP/RBAC mode only) not bearing additional privileges or security contexts interfering with existing security contexts.

The list of packages shown in Table 2-1 is identical for the LSPP/RBAC and CAPP modes of operation.

The list contains the packages with their version numbers for each architecture.

Table 2-1: List of TOE packages

	rpms-i386.txt	rpms-ia64.txt	rpms-x86_64.txt
Deployment_Guide-en-US	5.0.0.19	5.0.0.19	5.0.0.19
GConf2	2.14.0.9.el5	2.14.0.9.el5	2.14.0.9.el5
MAKEDEV	3.23.1.2	3.23.1.2	3.23.1.2
NetworkManager	0.6.4.6.el5	0.6.4.6.el5	0.6.4.6.el5
ORBit2	2.14.3.4.el5	2.14.3.4.el5	2.14.3.4.el5
OpenIPMI	2.0.6.5.el5.3	2.0.6.5.el5.3	2.0.6.5.el5.3
OpenIPMI-libs	2.0.6.5.el5.3	2.0.6.5.el5.3	2.0.6.5.el5.3
SysVinit	2.86.14	2.86.14	2.86.14
acl	2.2.39.2.1.el5	2.2.39.2.1.el5	2.2.39.2.1.el5
acpid	1.0.4.5	1.0.4.5	1.0.4.5
aide	0.12.9.el5	0.12.9.el5	0.12.9.el5
amtu	1.0.4.4	1.0.4.4	1.0.4.4
anacron	2.3.45.el5	2.3.45.el5	2.3.45.el5
apmd	3.2.2.5	-	-
aspell	0.60.3.7.1	0.60.3.7.1	0.60.3.7.1
aspell#2	-	-	0.60.3.7.1
aspell-en	6.0.2.1	6.0.2.1	6.0.2.1
at	3.1.8.82.fc6	3.1.8.82.fc6	3.1.8.82.fc6
atk	1.12.2.1.fc6	1.12.2.1.fc6	1.12.2.1.fc6
attr	2.4.32.1.1	2.4.32.1.1	2.4.32.1.1
audit	1.3.1.6.el5	1.3.1.6.el5	1.3.1.6.el5
audit-libs	1.3.1.6.el5	1.3.1.6.el5	1.3.1.6.el5
audit-libs#2	-	-	1.3.1.6.el5
audit-libs-devel	1.3.1.6.el5	1.3.1.6.el5	1.3.1.6.el5
audit-libs-devel#2	-	-	1.3.1.6.el5
audit-libs-python	1.3.1.6.el5	1.3.1.6.el5	1.3.1.6.el5
authconfig	5.3.12.2.el5	5.3.12.2.el5	5.3.12.2.el5
autoconf	2.59.12	2.59.12	2.59.12
autofs	5.0.1.0.rc2.42	5.0.1.0.rc2.42	5.0.1.0.rc2.42
automake	1.9.6.2.1	1.9.6.2.1	1.9.6.2.1
basesystem	8.0.5.1.1	8.0.5.1.1	8.0.5.1.1
bash	3.1.16.1	3.1.16.1	3.1.16.1
bc	1.06.21	1.06.21	1.06.21
beecrypt	4.1.2.10.1.1	4.1.2.10.1.1	4.1.2.10.1.1
bind-libs	9.3.3.7.el5	9.3.3.7.el5	9.3.3.7.el5
bind-utils	9.3.3.7.el5	9.3.3.7.el5	9.3.3.7.el5
binutils	2.17.50.0.6.2.el5	2.17.50.0.6.2.el5	2.17.50.0.6.2.el5
bison	2.3.2.1	2.3.2.1	2.3.2.1
bluez-gnome	0.5.5.fc6	0.5.5.fc6	0.5.5.fc6
bluez-libs	3.7.1	3.7.1	3.7.1
bluez-utils	3.7.2	3.7.2	3.7.2
bzip2	1.0.3.3	1.0.3.3	1.0.3.3
bzip2-libs	1.0.3.3	1.0.3.3	1.0.3.3
cairo	1.2.4.1.fc6	1.2.4.1.fc6	1.2.4.1.fc6
capp-lspp-eal4-config-hp	0.64.4	0.64.4	0.64.4
ccid	1.0.1.6.el5	1.0.1.6.el5	1.0.1.6.el5
checkpolicy	1.33.1.2.el5	1.33.1.2.el5	1.33.1.2.el5
chkconfig	1.3.30.1.1	1.3.30.1.1	1.3.30.1.1
chkfontpath	1.10.1.1.1	1.10.1.1.1	1.10.1.1.1
conman	0.1.9.2.4.el5	0.1.9.2.4.el5	0.1.9.2.4.el5
coolkey	1.0.1.16.el5	1.0.1.16.el5	1.0.1.16.el5
coolkey#2	-	-	1.0.1.16.el5
coreutils	5.97.12.1.el5	5.97.12.1.el5	5.97.12.1.el5
cpio	2.6.20	2.6.20	2.6.20
cpp	4.1.1.52.el5	4.1.1.52.el5	4.1.1.52.el5
cpuspeed	1.2.1.1.45.el5	1.2.1.1.45.el5	1.2.1.1.45.el5
cracklib	2.8.9.3.1	2.8.9.3.1	2.8.9.3.1
cracklib#2	-	-	2.8.9.3.1
cracklib-dicts	2.8.9.3.1	2.8.9.3.1	2.8.9.3.1
crash	4.0.3.14	4.0.3.14	4.0.3.14

crontabs	1.10.8	1.10.8	1.10.8
cryptsetup-luks	1.0.3.2.2.el5	1.0.3.2.2.el5	1.0.3.2.2.el5
cryptsetup-luks#2	-	-	1.0.3.2.2.el5
cups	1.2.4.11.8.el5	1.2.4.11.8.el5	1.2.4.11.8.el5
cups-libs	1.2.4.11.8.el5	1.2.4.11.8.el5	1.2.4.11.8.el5
cups-libs#2	-	-	1.2.4.11.8.el5
curl	7.15.5.2.el5	7.15.5.2.el5	7.15.5.2.el5
cvs	1.11.22.5.el5	1.11.22.5.el5	1.11.22.5.el5
cyrus-sasl	2.1.22.4	2.1.22.4	2.1.22.4
cyrus-sasl-devel	2.1.22.4	2.1.22.4	2.1.22.4
cyrus-sasl-lib	2.1.22.4	2.1.22.4	2.1.22.4
cyrus-sasl-lib#2	-	-	2.1.22.4
cyrus-sasl-plain	2.1.22.4	2.1.22.4	2.1.22.4
cyrus-sasl-plain#2	-	-	2.1.22.4
db4	4.3.29.9.fc6	4.3.29.9.fc6	4.3.29.9.fc6
db4#2	-	-	4.3.29.9.fc6
dbus	1.0.0.6.el5	1.0.0.6.el5	1.0.0.6.el5
dbus-glib	0.70.5	0.70.5	0.70.5
dbus-python	0.70.7.el5	0.70.7.el5	0.70.7.el5
desktop-file-utils	0.10.7	0.10.7	0.10.7
device-mapper	1.02.13.1.el5	1.02.13.1.el5	1.02.13.1.el5
device-mapper#2	-	-	1.02.13.1.el5
dhcdbd	2.2.1.el5	2.2.1.el5	2.2.1.el5
dhclient	3.0.5.3.el5	3.0.5.3.el5	3.0.5.3.el5
dhcpv6_client	0.10.33.el5	0.10.33.el5	0.10.33.el5
diffutils	2.8.1.15.2.2	2.8.1.15.2.2	2.8.1.15.2.2
dmidecode	2.7.1.28.2.el5	-	2.7.1.28.2.el5
dmraid	1.0.0.rc13.2.el5	1.0.0.rc13.2.el5	1.0.0.rc13.2.el5
dos2unix	3.1.27.1	3.1.27.1	3.1.27.1
dosfstools	2.11.6.2.el5	2.11.6.2.el5	2.11.6.2.el5
dump	0.4b41.2.fc6	0.4b41.2.fc6	0.4b41.2.fc6
e2fsprogs	1.39.8.el5	1.39.8.el5	1.39.8.el5
e2fsprogs-devel	1.39.8.el5	1.39.8.el5	1.39.8.el5
e2fsprogs-libs	1.39.8.el5	1.39.8.el5	1.39.8.el5
e2fsprogs-libs#2	-	-	1.39.8.el5
ed	0.2.38.2.2	0.2.38.2.2	0.2.38.2.2
eject	2.1.5.4.2.el5	2.1.5.4.2.el5	2.1.5.4.2.el5
elfutils	0.125.3.el5	0.125.3.el5	0.125.3.el5
elfutils-libelf	0.125.3.el5	0.125.3.el5	0.125.3.el5
elfutils-libs	0.125.3.el5	0.125.3.el5	0.125.3.el5
elilo	-	3.6.2	-
elinks	0.11.1.5.1.el5	0.11.1.5.1.el5	0.11.1.5.1.el5
ethtool	5.1.el5	5.1.el5	5.1.el5
expat	1.95.8.8.2.1	1.95.8.8.2.1	1.95.8.8.2.1
expat#2	-	-	1.95.8.8.2.1
expect	5.43.0.5.1	5.43.0.5.1	5.43.0.5.1
expect#2	-	-	5.43.0.5.1
expect-devel	5.43.0.5.1	5.43.0.5.1	5.43.0.5.1
expect-devel#2	-	-	5.43.0.5.1
fbset	2.1.22	2.1.22	2.1.22
file	4.17.8	4.17.8	4.17.8
filesystem	2.4.0.1	2.4.0.1	2.4.0.1
findutils	4.2.27.4.1	4.2.27.4.1	4.2.27.4.1
finger	0.17.32.2.1.1	0.17.32.2.1.1	0.17.32.2.1.1
firstboot-tui	1.4.27.2.1.el5	1.4.27.2.1.el5	1.4.27.2.1.el5
flex	2.5.4a.41.fc6	2.5.4a.41.fc6	2.5.4a.41.fc6
fontconfig	2.4.1.6.el5	2.4.1.6.el5	2.4.1.6.el5
freetype	2.2.1.16.el5	2.2.1.16.el5	2.2.1.16.el5
ftp	0.17.33.fc6	0.17.33.fc6	0.17.33.fc6
gawk	3.1.5.14.el5	3.1.5.14.el5	3.1.5.14.el5
gcc	4.1.1.52.el5	4.1.1.52.el5	4.1.1.52.el5
gcc-c++	4.1.1.52.el5	4.1.1.52.el5	4.1.1.52.el5
gdbm	1.8.0.26.2.1	1.8.0.26.2.1	1.8.0.26.2.1
gettext	0.14.6.4.el5	0.14.6.4.el5	0.14.6.4.el5
ghostscript	8.15.2.9.1.el5	8.15.2.9.1.el5	8.15.2.9.1.el5
ghostscript#2	-	-	8.15.2.9.1.el5
glib2	2.12.3.2.fc6	2.12.3.2.fc6	2.12.3.2.fc6
glib2-devel	2.12.3.2.fc6	2.12.3.2.fc6	2.12.3.2.fc6
glibc	2.5.12	2.5.12	2.5.12
glibc#2	-	-	2.5.12
glibc-common	2.5.12	2.5.12	2.5.12
glibc-devel	2.5.12	2.5.12	2.5.12
glibc-devel#2	-	-	2.5.12
glibc-headers	2.5.12	2.5.12	2.5.12
gnu-efi	3.0c.1.1	3.0c.1.1	-
gnupg	1.4.5.12	1.4.5.12	1.4.5.12
gnutls	1.4.1.2	1.4.1.2	1.4.1.2
gnutls#2	-	-	1.4.1.2

gpg-pubkey	37017186.45761324	37017186.45761324	37017186.45761324
gpm	1.20.1.74.1	1.20.1.74.1	1.20.1.74.1
gpm#2	-	-	1.20.1.74.1
grep	2.5.1.54.2.el5	2.5.1.54.2.el5	2.5.1.54.2.el5
groff	1.18.1.1.11.1	1.18.1.1.11.1	1.18.1.1.11.1
grub	0.97.13	-	0.97.13
gtk2	2.10.4.16.el5	2.10.4.16.el5	2.10.4.16.el5
gzip	1.3.5.9.el5	1.3.5.9.el5	1.3.5.9.el5
hal	0.5.8.1.19.el5	0.5.8.1.19.el5	0.5.8.1.19.el5
hdparm	6.6.2	6.6.2	6.6.2
hesiod	3.1.0.8	3.1.0.8	3.1.0.8
htmlview	4.0.0.1.el5	4.0.0.1.el5	4.0.0.1.el5
hwdata	0.194.1	0.194.1	0.194.1
ifd-egate	0.05.15	0.05.15	0.05.15
imake	1.0.2.3	1.0.2.3	1.0.2.3
info	4.8.14.el5	4.8.14.el5	4.8.14.el5
initscripts	8.45.14.EL.1	8.45.14.EL.1	8.45.14.EL.1
iproute	2.6.18.4.el5	2.6.18.4.el5	2.6.18.4.el5
ipsec-tools	0.6.5.8.el5	0.6.5.8.el5	0.6.5.8.el5
iptables	1.3.5.1.2.1	1.3.5.1.2.1	1.3.5.1.2.1
iptables-ipv6	1.3.5.1.2.1	1.3.5.1.2.1	1.3.5.1.2.1
iptstate	1.4.1.1.2.2	1.4.1.1.2.2	1.4.1.1.2.2
iputils	20020927.43.el5	20020927.43.el5	20020927.43.el5
irda-utils	0.9.17.2.fc6	0.9.17.2.fc6	0.9.17.2.fc6
irqbalance	1.13.9.el5	1.13.9.el5	1.13.9.el5
jwhois	3.2.3.8.el5	3.2.3.8.el5	3.2.3.8.el5
kbd	1.12.19.el5	1.12.19.el5	1.12.19.el5
kernel	2.6.18.8.1.3.lspp.81.el5	2.6.18.8.1.3.lspp.81.el5	
2.6.18.8.1.3.lspp.81.el5			
kernel-devel	2.6.18.8.1.3.lspp.81.el5	2.6.18.8.1.3.lspp.81.el5	
2.6.18.8.1.3.lspp.81.el5			
kernel-headers	2.6.18.8.el5	2.6.18.8.el5	2.6.18.8.el5
keyutils-libs	1.2.1.el5	1.2.1.el5	1.2.1.el5
keyutils-libs-devel	1.2.1.el5	1.2.1.el5	1.2.1.el5
keyutils-libs-devel#2	-	-	1.2.1.el5
kpartx	0.4.7.8.el5	0.4.7.8.el5	0.4.7.8.el5
krb5-devel	1.5.17	1.5.17	1.5.17
krb5-libs	1.5.17	1.5.17	1.5.17
krb5-libs#2	-	-	1.5.17
krb5-workstation	1.5.17	1.5.17	1.5.17
ksh	20060214.1.4	20060214.1.4	20060214.1.4
kudzu	1.2.57.1.13.1	1.2.57.1.13.1	1.2.57.1.13.1
less	394.5.el5	394.5.el5	394.5.el5
lftp	3.5.1.2.fc6	3.5.1.2.fc6	3.5.1.2.fc6
libFS	1.0.0.3.1	1.0.0.3.1	1.0.0.3.1
libICE	1.0.1.2.1	1.0.1.2.1	1.0.1.2.1
libICE#2	-	-	1.0.1.2.1
libIDL	0.8.7.1.fc6	0.8.7.1.fc6	0.8.7.1.fc6
libSM	1.0.1.3.1	1.0.1.3.1	1.0.1.3.1
libSM#2	-	-	1.0.1.3.1
libX11	1.0.3.8.el5	1.0.3.8.el5	1.0.3.8.el5
libX11#2	-	-	1.0.3.8.el5
libXau	1.0.1.3.1	1.0.1.3.1	1.0.1.3.1
libXau#2	-	-	1.0.1.3.1
libXcursor	1.1.7.1.1	1.1.7.1.1	1.1.7.1.1
libXdmcp	1.0.1.2.1	1.0.1.2.1	1.0.1.2.1
libXdmcp#2	-	-	1.0.1.2.1
libXext	1.0.1.2.1	1.0.1.2.1	1.0.1.2.1
libXext#2	-	-	1.0.1.2.1
libXfixes	4.0.1.2.1	4.0.1.2.1	4.0.1.2.1
libXfont	1.2.2.1.fc6	1.2.2.1.fc6	1.2.2.1.fc6
libXft	2.1.10.1.1	2.1.10.1.1	2.1.10.1.1
libXi	1.0.1.3.1	1.0.1.3.1	1.0.1.3.1
libXi#2	-	-	1.0.1.3.1
libXinerama	1.0.1.2.1	1.0.1.2.1	1.0.1.2.1
libXrandr	1.1.1.3.1	1.1.1.3.1	1.1.1.3.1
libXrender	0.9.1.3.1	0.9.1.3.1	0.9.1.3.1
libXres	1.0.1.3.1	1.0.1.3.1	1.0.1.3.1
libXt	1.0.2.3.1.fc6	1.0.2.3.1.fc6	1.0.2.3.1.fc6
libXt#2	-	-	1.0.2.3.1.fc6
libXxf86vm	1.0.1.3.1	1.0.1.3.1	1.0.1.3.1
libXxf86vm#2	-	-	1.0.1.3.1
libacl	2.2.39.2.1.el5	2.2.39.2.1.el5	2.2.39.2.1.el5
libacl#2	-	-	2.2.39.2.1.el5
libacl-devel	2.2.39.2.1.el5	2.2.39.2.1.el5	2.2.39.2.1.el5
libacl-devel#2	-	-	2.2.39.2.1.el5
libaio	0.3.106.3.2	0.3.106.3.2	0.3.106.3.2
libaio#2	-	-	0.3.106.3.2

libattr	2.4.32.1.1	2.4.32.1.1	2.4.32.1.1
libattr#2	-	-	2.4.32.1.1
libattr-devel	2.4.32.1.1	2.4.32.1.1	2.4.32.1.1
libattr-devel#2	-	-	2.4.32.1.1
libcap	1.10.26	1.10.26	1.10.26
libcap#2	-	-	1.10.26
libcap-devel	1.10.26	1.10.26	1.10.26
libcap-devel#2	-	-	1.10.26
libdrm	2.0.2.1.1	2.0.2.1.1	2.0.2.1.1
libdrm#2	-	-	2.0.2.1.1
libevent	1.1a.3.2.1	1.1a.3.2.1	1.1a.3.2.1
libfontenc	1.0.2.2.2.el5	1.0.2.2.2.el5	1.0.2.2.2.el5
libgcc	4.1.1.52.el5	4.1.1.52.el5	4.1.1.52.el5
libgcc#2	-	-	4.1.1.52.el5
libgcrypt	1.2.3.1	1.2.3.1	1.2.3.1
libgcrypt#2	-	-	1.2.3.1
libgomp	4.1.1.52.el5	4.1.1.52.el5	4.1.1.52.el5
libgpg-error	1.4.2	1.4.2	1.4.2
libgpg-error#2	-	-	1.4.2
libgssapi	0.10.2	0.10.2	0.10.2
libhugetlbfs	1.0.1.1.el5	-	1.0.1.1.el5
libhugetlbfs-lib	1.0.1.1.el5	-	1.0.1.1.el5
libidn	0.6.5.1.1	0.6.5.1.1	0.6.5.1.1
libjpeg	6b.37	6b.37	6b.37
libjpeg#2	-	-	6b.37
libnl	1.0.0.10.pre5.4	1.0.0.10.pre5.4	1.0.0.10.pre5.4
libnotify	0.4.2.6.el5	0.4.2.6.el5	0.4.2.6.el5
libpcap	0.9.4.8.1	0.9.4.8.1	0.9.4.8.1
libpng	1.2.10.7	1.2.10.7	1.2.10.7
libpng#2	-	-	1.2.10.7
libselinux	1.33.4.4.el5	1.33.4.4.el5	1.33.4.4.el5
libselinux#2	-	-	1.33.4.4.el5
libselinux-devel	1.33.4.4.el5	1.33.4.4.el5	1.33.4.4.el5
libselinux-python	1.33.4.4.el5	1.33.4.4.el5	1.33.4.4.el5
libsemanage	1.9.1.3.el5	1.9.1.3.el5	1.9.1.3.el5
libsemanage-devel	1.9.1.3.el5	1.9.1.3.el5	1.9.1.3.el5
libsepol	1.15.2.1.el5	1.15.2.1.el5	1.15.2.1.el5
libsepol#2	-	-	1.15.2.1.el5
libsepol-devel	1.15.2.1.el5	1.15.2.1.el5	1.15.2.1.el5
libstdc++	4.1.1.52.el5	4.1.1.52.el5	4.1.1.52.el5
libstdc++#2	-	-	4.1.1.52.el5
libstdc++-devel	4.1.1.52.el5	4.1.1.52.el5	4.1.1.52.el5
libsysfs	2.0.0.6	2.0.0.6	2.0.0.6
libtermcap	2.0.8.46.1	2.0.8.46.1	2.0.8.46.1
libtermcap#2	-	-	2.0.8.46.1
libtermcap-devel	2.0.8.46.1	2.0.8.46.1	2.0.8.46.1
libtiff	3.8.2.7.el5	3.8.2.7.el5	3.8.2.7.el5
libtiff#2	-	-	3.8.2.7.el5
libusb	0.1.12.5.1	0.1.12.5.1	0.1.12.5.1
libuser	0.54.7.2.el5.1	0.54.7.2.el5.1	0.54.7.2.el5.1
libuser-devel	0.54.7.2.el5.1	0.54.7.2.el5.1	0.54.7.2.el5.1
libutempter	1.1.4.3.fc6	1.1.4.3.fc6	1.1.4.3.fc6
libutempter#2	-	-	1.1.4.3.fc6
libvolume_id	095.14.5.el5	095.14.5.el5	095.14.5.el5
libwnck	2.16.0.4.fc6	2.16.0.4.fc6	2.16.0.4.fc6
libxml2	2.6.26.2.1.2	2.6.26.2.1.2	2.6.26.2.1.2
libxml2-python	2.6.26.2.1.2	2.6.26.2.1.2	2.6.26.2.1.2
logrotate	3.7.4.7	3.7.4.7	3.7.4.7
logwatch	7.3.5	7.3.5	7.3.5
lsof	4.78.3	4.78.3	4.78.3
lvm2	2.02.16.3.el5	2.02.16.3.el5	2.02.16.3.el5
m2crypto	0.16.6.el5.1	0.16.6.el5.1	0.16.6.el5.1
m4	1.4.5.3.el5.1	1.4.5.3.el5.1	1.4.5.3.el5.1
mailcap	2.1.23.1.fc6	2.1.23.1.fc6	2.1.23.1.fc6
mailx	8.1.1.44.2.2	8.1.1.44.2.2	8.1.1.44.2.2
make	3.81.1.1	3.81.1.1	3.81.1.1
man	1.6d.1.1	1.6d.1.1	1.6d.1.1
man-pages	2.39.9.el5	2.39.9.el5	2.39.9.el5
mcelog	-	-	0.7.1.22.fc6
mcstrans	0.2.3.1.el5	0.2.3.1.el5	0.2.3.1.el5
mdadm	2.5.4.3.el5	2.5.4.3.el5	2.5.4.3.el5
mesa-libGL	6.5.1.7.2.el5	6.5.1.7.2.el5	6.5.1.7.2.el5
mesa-libGL#2	-	-	6.5.1.7.2.el5
mgetty	1.1.33.9.fc6	1.1.33.9.fc6	1.1.33.9.fc6
microcode_ctl	1.15.1.40.el5	-	1.15.1.40.el5
mingetty	1.07.5.2.2	1.07.5.2.2	1.07.5.2.2
mkbootdisk	1.5.3.2.1	-	1.5.3.2.1
mkinitrd	5.1.19.6.1	5.1.19.6.1	5.1.19.6.1

mkinitrd#2	-	-	5.1.19.6.1
mktemp	1.5.23.2.2	1.5.23.2.2	1.5.23.2.2
mlocate	0.15.1.el5	0.15.1.el5	0.15.1.el5
module-init-tools	3.3.0.pre3.1.16.el5	3.3.0.pre3.1.16.el5	
3.3.0.pre3.1.16.el5			
mtools	3.9.10.2.fc6	3.9.10.2.fc6	3.9.10.2.fc6
mtr	0.71.3.1	0.71.3.1	0.71.3.1
nano	1.3.12.1.1	1.3.12.1.1	1.3.12.1.1
nash	5.1.19.6.1	5.1.19.6.1	5.1.19.6.1
nc	1.84.10.fc6	1.84.10.fc6	1.84.10.fc6
ncurses	5.5.24.20060715	5.5.24.20060715	5.5.24.20060715
ncurses#2	-	-	5.5.24.20060715
net-snmp-libs	5.3.1.14.el5	5.3.1.14.el5	5.3.1.14.el5
net-tools	1.60.73	1.60.73	1.60.73
netlabel_tools	0.17.9.el5	0.17.9.el5	0.17.9.el5
newt	0.52.2.9	0.52.2.9	0.52.2.9
nfs-utils	1.0.9.16.el5	1.0.9.16.el5	1.0.9.16.el5
nfs-utils-lib	1.0.8.7.2	1.0.8.7.2	1.0.8.7.2
notification-daemon	0.3.5.8.el5	0.3.5.8.el5	0.3.5.8.el5
nscd	2.5.12	2.5.12	2.5.12
nspr	4.6.5.1.el5	4.6.5.1.el5	4.6.5.1.el5
nspr#2	-	-	4.6.5.1.el5
nss	3.11.5.1.el5	3.11.5.1.el5	3.11.5.1.el5
nss#2	-	-	3.11.5.1.el5
nss-tools	3.11.5.1.el5	3.11.5.1.el5	3.11.5.1.el5
nss_db	2.2.35.1	2.2.35.1	2.2.35.1
nss_db#2	-	-	2.2.35.1
nss_ldap	253.3	253.3	253.3
nss_ldap#2	-	-	253.3
ntsysv	1.3.30.1.1	1.3.30.1.1	1.3.30.1.1
numactl	0.9.8.2.el5	0.9.8.2.el5	0.9.8.2.el5
numactl#2	-	-	0.9.8.2.el5
openldap	2.3.27.5	2.3.27.5	2.3.27.5
openldap#2	-	-	2.3.27.5
openssh	4.3p2.21.el5	4.3p2.21.el5	4.3p2.21.el5
openssh-clients	4.3p2.21.el5	4.3p2.21.el5	4.3p2.21.el5
openssh-server	4.3p2.21.el5	4.3p2.21.el5	4.3p2.21.el5
openssl	0.9.8b.8.3.el5	0.9.8b.8.3.el5	0.9.8b.8.3.el5
openssl#2	-	-	0.9.8b.8.3.el5
openssl-devel	0.9.8b.8.3.el5	0.9.8b.8.3.el5	0.9.8b.8.3.el5
pam	0.99.6.2.3.22.el5	0.99.6.2.3.22.el5	0.99.6.2.3.22.el5
pam#2	-	-	0.99.6.2.3.22.el5
pam-devel	0.99.6.2.3.22.el5	0.99.6.2.3.22.el5	0.99.6.2.3.22.el5
pam_ccreds	3.5	3.5	3.5
pam_ccreds#2	-	-	3.5
pam_krb5	2.2.11.1	2.2.11.1	2.2.11.1
pam_krb5#2	-	-	2.2.11.1
pam_passwdqc	1.0.2.1.2.2	1.0.2.1.2.2	1.0.2.1.2.2
pam_passwdqc#2	-	-	1.0.2.1.2.2
pam_pkcs11	0.5.3.23	0.5.3.23	0.5.3.23
pam_pkcs11#2	-	-	0.5.3.23
pam_smb	1.1.7.7.2.1	1.1.7.7.2.1	1.1.7.7.2.1
pam_smb#2	-	-	1.1.7.7.2.1
pango	1.14.9.3.el5	1.14.9.3.el5	1.14.9.3.el5
paps	0.6.6.17.el5	0.6.6.17.el5	0.6.6.17.el5
parted	1.8.1.4.el5	1.8.1.4.el5	1.8.1.4.el5
parted#2	-	-	1.8.1.4.el5
passwd	0.73.1	0.73.1	0.73.1
patch	2.5.4.29.2.2	2.5.4.29.2.2	2.5.4.29.2.2
pax	3.4.1.2.2	3.4.1.2.2	3.4.1.2.2
pciutils	2.2.3.4	2.2.3.4	2.2.3.4
pciutils-devel	2.2.3.4	2.2.3.4	2.2.3.4
pciutils-devel#2	-	-	2.2.3.4
pcmciautils	014.5	014.5	014.5
pcre	6.6.1.1	6.6.1.1	6.6.1.1
pcsc-lite	1.3.1.7	1.3.1.7	1.3.1.7
pcsc-lite-libs	1.3.1.7	1.3.1.7	1.3.1.7
perl	5.8.8.10	5.8.8.10	5.8.8.10
perl-Digest-HMAC	1.01.15	1.01.15	1.01.15
perl-Digest-SHA1	2.11.1.2.1	2.11.1.2.1	2.11.1.2.1
perl-String-CRC32	1.4.2.fc6	1.4.2.fc6	1.4.2.fc6
pinfo	0.6.9.1.fc6	0.6.9.1.fc6	0.6.9.1.fc6
pkgconfig	0.21.1.fc6	0.21.1.fc6	0.21.1.fc6
pkinit-nss	0.3.5.1.el5	0.3.5.1.el5	0.3.5.1.el5
pm-utils	0.19.3	0.19.3	0.19.3
policycoreutils	1.33.12.8.el5	1.33.12.8.el5	1.33.12.8.el5
policycoreutils-newrole	1.33.12.8.el5	1.33.12.8.el5	1.33.12.8.el5
popt	1.10.2.37.el5	1.10.2.37.el5	1.10.2.37.el5

portmap	4.0.65.2.2.1	4.0.65.2.2.1	4.0.65.2.2.1
postfix	2.3.3.2	2.3.3.2	2.3.3.2
ppp	2.4.4.1.el5	2.4.4.1.el5	2.4.4.1.el5
prctl	-	1.4.5.2.1	-
prelink	0.3.9.2	-	0.3.9.2
procps	3.2.7.8.1.el5	3.2.7.8.1.el5	3.2.7.8.1.el5
psacct	6.3.2.41.1	6.3.2.41.1	6.3.2.41.1
psmisc	22.2.5	22.2.5	22.2.5
pyOpenSSL	0.6.1.p24.7.2.2	0.6.1.p24.7.2.2	0.6.1.p24.7.2.2
pygobject2	2.12.1.5.el5	2.12.1.5.el5	2.12.1.5.el5
python	2.4.3.19.el5	2.4.3.19.el5	2.4.3.19.el5
python-devel	2.4.3.19.el5	2.4.3.19.el5	2.4.3.19.el5
python-devel#2	-	-	2.4.3.19.el5
python-elementtree	1.2.6.5	1.2.6.5	1.2.6.5
python-sqlite	1.1.7.1.2.1	1.1.7.1.2.1	1.1.7.1.2.1
python-urlgrabber	3.1.0.2	3.1.0.2	3.1.0.2
quota	3.13.1.2.3.2.el5	3.13.1.2.3.2.el5	3.13.1.2.3.2.el5
rdate	1.4.6	1.4.6	1.4.6
rdist	6.1.5.44	6.1.5.44	6.1.5.44
readahead	1.3.7.el5	1.3.7.el5	1.3.7.el5
readline	5.1.1.1	5.1.1.1	5.1.1.1
readline#2	-	-	5.1.1.1
readline-devel	5.1.1.1	5.1.1.1	5.1.1.1
readline-devel#2	-	-	5.1.1.1
redhat-logos	4.9.16.1	4.9.16.1	4.9.16.1
redhat-lsb	3.1.12.2.EL	3.1.12.2.EL	3.1.12.2.EL
redhat-lsb#2	-	-	3.1.12.2.EL
redhat-menus	6.7.8.1.el5	6.7.8.1.el5	6.7.8.1.el5
redhat-release	5Server.5.0.0.9	5Server.5.0.0.9	5Server.5.0.0.9
redhat-release-notes	5Server.5	5Server.5	5Server.5
rhel-instnum	1.0.7.1.el5	1.0.7.1.el5	1.0.7.1.el5
rhn-check	0.4.13.1.el5	0.4.13.1.el5	0.4.13.1.el5
rhn-client-tools	0.4.13.1.el5	0.4.13.1.el5	0.4.13.1.el5
rhn-setup	0.4.13.1.el5	0.4.13.1.el5	0.4.13.1.el5
rhnlib	2.2.5.1.el5	2.2.5.1.el5	2.2.5.1.el5
rhnsd	4.6.1.1.el5	4.6.1.1.el5	4.6.1.1.el5
rhpl	0.194.1.1	0.194.1.1	0.194.1.1
rmt	0.4b41.2.fc6	0.4b41.2.fc6	0.4b41.2.fc6
rng-utils	2.0.1.14.1.fc6	2.0.1.14.1.fc6	2.0.1.14.1.fc6
rootfiles	8.1.1.1.1	8.1.1.1.1	8.1.1.1.1
rp-pppoe	3.5.32.1	3.5.32.1	3.5.32.1
rpm	4.4.2.37.el5	4.4.2.37.el5	4.4.2.37.el5
rpm-build	4.4.2.37.el5	4.4.2.37.el5	4.4.2.37.el5
rpm-libs	4.4.2.37.el5	4.4.2.37.el5	4.4.2.37.el5
rpm-python	4.4.2.37.el5	4.4.2.37.el5	4.4.2.37.el5
rsh	0.17.37.el5	0.17.37.el5	0.17.37.el5
rsync	2.6.8.3.1	2.6.8.3.1	2.6.8.3.1
salinfo	-	1.1.2.el5	-
sed	4.1.5.5.fc6	4.1.5.5.fc6	4.1.5.5.fc6
selinux-policy	2.4.6.67.el5	2.4.6.67.el5	2.4.6.67.el5
selinux-policy-devel	2.4.6.67.el5	2.4.6.67.el5	2.4.6.67.el5
selinux-policy-mls	2.4.6.67.el5	2.4.6.67.el5	2.4.6.67.el5
selinux-policy-strict	2.4.6.67.el5	2.4.6.67.el5	2.4.6.67.el5
selinux-policy-targeted	2.4.6.67.el5	2.4.6.67.el5	2.4.6.67.el5
setarch	2.0.1.1	2.0.1.1	2.0.1.1
setools	3.0.3.el5	3.0.3.el5	3.0.3.el5
setserial	2.17.19.2.2	2.17.19.2.2	2.17.19.2.2
setup	2.5.58.1.el5	2.5.58.1.el5	2.5.58.1.el5
setuptool	1.19.2.1	1.19.2.1	1.19.2.1
shadow-utils	4.0.17.12.el5	4.0.17.12.el5	4.0.17.12.el5
slang	2.0.6.4.el5	2.0.6.4.el5	2.0.6.4.el5
smartmontools	5.36.3.1.el5	5.36.3.1.el5	5.36.3.1.el5
sos	1.3.1.el5	1.3.1.el5	1.3.1.el5
specspo	13.1.el5	13.1.el5	13.1.el5
sqlite	3.3.6.2	3.3.6.2	3.3.6.2
star	1.5a75.1	1.5a75.1	1.5a75.1
startup-notification	0.8.4.1	0.8.4.1	0.8.4.1
strace	4.5.15.1.el5	4.5.15.1.el5	4.5.15.1.el5
stunnel	4.15.2	4.15.2	4.15.2
sudo	1.6.8p12.10	1.6.8p12.10	1.6.8p12.10
swig	1.3.29.2.el5	1.3.29.2.el5	1.3.29.2.el5
symlinks	1.2.24.2.2	1.2.24.2.2	1.2.24.2.2
sysfsutils	2.0.0.6	2.0.0.6	2.0.0.6
sysklogd	1.4.1.39.2	1.4.1.39.2	1.4.1.39.2
syslinux	3.11.4	-	3.11.4
sysreport	1.4.3.10.el5	1.4.3.10.el5	1.4.3.10.el5
system-config-network-tui	1.3.99.1.el5	1.3.99.1.el5	1.3.99.1.el5
system-config-securitylevel-tui	1.6.29.1.1.el5	1.6.29.1.1.el5	1.6.29.1.1.el5

talk	0.17.29.2.2	0.17.29.2.2	0.17.29.2.2
tar	1.15.1.23.el5	1.15.1.23.el5	1.15.1.23.el5
tcl	8.4.13.3.fc6	8.4.13.3.fc6	8.4.13.3.fc6
tcl#2	-	-	8.4.13.3.fc6
tcp_wrappers	7.6.40.2.1	7.6.40.2.1	7.6.40.2.1
tcp_wrappers#2	-	-	7.6.40.2.1
tcpdump	3.9.4.8.1	3.9.4.8.1	3.9.4.8.1
tcsh	6.14.12.el5	6.14.12.el5	6.14.12.el5
telnet	0.17.38.el5	0.17.38.el5	0.17.38.el5
termcap	5.5.1.20060701.1	5.5.1.20060701.1	5.5.1.20060701.1
texinfo	4.8.14.el5	4.8.14.el5	4.8.14.el5
time	1.7.27.2.2	1.7.27.2.2	1.7.27.2.2
tk	8.4.13.3.fc6	8.4.13.3.fc6	8.4.13.3.fc6
tk#2	-	-	8.4.13.3.fc6
tmpwatch	2.9.7.1.1.el5.1	2.9.7.1.1.el5.1	2.9.7.1.1.el5.1
traceroute	2.0.1.2.el5	2.0.1.2.el5	2.0.1.2.el5
tree	1.5.0.4	1.5.0.4	1.5.0.4
ttmkfdir	3.0.9.23.el5	3.0.9.23.el5	3.0.9.23.el5
tzdata	2006m.2.fc6	2006m.2.fc6	2006m.2.fc6
udev	095.14.5.el5	095.14.5.el5	095.14.5.el5
unix2dos	2.2.26.2.2	2.2.26.2.2	2.2.26.2.2
unzip	5.52.2.2.1	5.52.2.2.1	5.52.2.2.1
urw-fonts	2.3.6.1.1	2.3.6.1.1	2.3.6.1.1
usbutils	0.71.2.1	0.71.2.1	0.71.2.1
usermode	1.88.3.el5	1.88.3.el5	1.88.3.el5
util-linux	2.13.0.44.el5	2.13.0.44.el5	2.13.0.44.el5
vconfig	1.9.2.1	1.9.2.1	1.9.2.1
vim-minimal	7.0.109.3	7.0.109.3	7.0.109.3
vixie-cron	4.1.68.el5	4.1.68.el5	4.1.68.el5
vsftpd	2.0.5.10.el5	2.0.5.10.el5	2.0.5.10.el5
wget	1.10.2.7.el5	1.10.2.7.el5	1.10.2.7.el5
which	2.16.7	2.16.7	2.16.7
wireless-tools	28.2.el5	28.2.el5	28.2.el5
wireless-tools#2	-	-	28.2.el5
words	3.0.9	3.0.9	3.0.9
wpa_supplicant	0.4.8.10.1.fc6	0.4.8.10.1.fc6	0.4.8.10.1.fc6
xinetd	2.3.14.10.el5	2.3.14.10.el5	2.3.14.10.el5
xorg-x11-filesystem	7.1.2.fc6	7.1.2.fc6	7.1.2.fc6
xorg-x11-font-utils	7.1.2	7.1.2	7.1.2
xorg-x11-xfs	1.0.2.3.1	1.0.2.3.1	1.0.2.3.1
yp-tools	2.9.0.1	2.9.0.1	2.9.0.1
ypbind	1.19.7.el5	1.19.7.el5	1.19.7.el5
yum	3.0.1.5.el5	3.0.1.5.el5	3.0.1.5.el5
yum-metadata-parser	1.0.8.fc6	1.0.8.fc6	1.0.8.fc6
yum-rhn-plugin	0.4.3.1.el5	0.4.3.1.el5	0.4.3.1.el5
yum-updatesd	3.0.1.5.el5	3.0.1.5.el5	3.0.1.5.el5
zip	2.31.1.2.2	2.31.1.2.2	2.31.1.2.2
zlib	1.2.3.3	1.2.3.3	1.2.3.3
zlib#2	-	-	1.2.3.3
zlib-devel	1.2.3.3	1.2.3.3	1.2.3.3
zlib-devel#2	-	-	1.2.3.3

The following remarks need to be considered when reading the table above:

- The x86_64 systems support execution of 64bit and 32bit binaries. To support this, some packages are installed and listed twice, once for each word size, using the ”#2” suffix to the package name for the second copy of the package. The suffix is not part of the package name.
- The “redhat-release” package has a different package name for the “Client” and “Server” flavors of the RHEL product. This is informational data only and has no impact on security functionality.

2.4 Configurations

The evaluated configurations are defined as follows.

- The CC evaluated package set must be selected at install time in accordance with the description provided in the Evaluated Configuration Guide and installed accordingly.

- Red Hat Enterprise Linux supports the use of IPv4 and IPv6. Only the functional equivalent of IPv6 to IPv4 is covered in this evaluation, additional mechanisms defined for IPv6 like multicast or IPSEC encryption is not tested.
- Both installation from CD-ROM or DVD-ROM and installation from a defined disk partition are supported.
- The default configuration for identification and authentication are the defined password based PAM modules. Support for other authentication options e.g. smartcard authentication, is not included in the evaluation configuration.
- If the system console is used, it must be connected directly to the system and afforded the same physical protection as the server.
- The chosen mode of operation (LSPP/RBAC or CAPP) must be configured by following the instructions pertaining to the respective mode of operation in the Evaluated Confuguration Guide.

The TOE comprises a single server machine (and optional peripherals) listed in section 2.4.2 running the system software listed the package list in section 2.3 (a server running the above listed software is referred to as a “TOE server” below).

Several TOE servers may be interlinked in a network, and individual networks may be joined by bridges and/or routers, or by TOE servers which act as routers and/or gateways. Each of the TOE servers implements its own security policy. The TOE does not include any synchronization function for those policies. As a result a single user may have user accounts on each of those servers with different user IDs, different roles, and other different attributes. (A synchronization method may optionally be used, but it not part of the TOE and must not use methods that conflict with the TOE requirements.)If other systems are connected to a network they need to be configured and managed by the same authority using an appropriate security policy that does not conflict with the security policy of the TOE. All links between this network and untrusted networks (e. g. the Internet) need to be protected by appropriate measures such as carefully configured firewall systems that prohibit attacks from the untrusted networks. Those protections are part of the TOE environment.

2.4.1 File systems

The following file system types are supported:

- Ext3 journaling filesystem,
- VFAT filesystem for the /boot/efi partition on Itanium systems,
- the ISO 9660 filesystem for CD-ROM drives,
- the ISO 9660 filesystem for DVD-ROM drives,
- The process file system, procfs (/proc), provides access to the process image of each process on the machine as if the process were a “file”. Process access decisions are enforced by DAC attributes inferred from the underlying process’ DAC attributes. Additional restrictions apply for specific objects in this file system.
- The sysfs filesystem (sysfs) used to export and handle non-process related kernel information such as driver specific information.
- The temporary filesystem (tmpfs) used as a temporary RAM based file system.
- The pseudoterminal device file system (devpts) used to provide pseudo terminal support.
- The virtual root file system (rootfs) used temporarily during system startup.
- The miscellaneous binary file format registration file system (binfmt_misc) used to configure interpreters for executing binary files based on file header information.
- The Security Enhanced Linux file system (selinuxfs) used for configuring the selinux system.

2.4.2 TOE hardware

The hardware on which the software components of the TOE are executed is considered part of the TOE.

The TOE includes each of the following hardware platforms.

- HP Intel Itanium2 (single and multi-core) processor based servers:
  - HP Integrity Superdome product line
  - HP Integrity rx product line
  - HP Integrity cx product line
  - HP Integrity BL product line
- Intel Xeon based servers with EM64T 64bit extensions (single and multi-core), and HP AMD Opteron processor (single and multi-core):
  - HP Proliant ML product line (EM64T capable models)
  - HP Proliant DL product line (EM64T capable or Opteron models)
  - HP ProLiant BL product line (EM64T capable or Opteron models)
- HP Intel Pentium and Xeon processor based servers without EM64T extensions:
  - HP Proliant ML product line (except EM64T capable models)
  - HP Proliant DL product line (except EM64T capable or Opteron models)
  - HP Proliant BL product line (except EM64T capable or Opteron models)
- HP Intel Xeon processor based systems:
  - HP xw product line
- HP Intel Pentium 4 processor based systems:
  - HP xw product line
  - HP Compaq dc series product line

The hardware that is allowed to be used for the evaluated configuration consists of the above listed platforms. However, it is not permitted to install the TOE within a nPar hardware partition.

All network adapters supported by the TOE are part of the TOE.

2.4.2.1 Peripherials

The following types of printers in the CAPP mode are supported:

- Printers supporting PCL version 4 (parallel, USB and Ethernet)
- Printers supporting PostScript level 1 (parallel, USB and Ethernet)

The following types of printers in the LSPP/RBAC mode are supported:

- Printers supporting PCL version 4 (parallel, USB)
- Printers supporting PostScript level 1 (parallel, USB)

Only printers supporting the above mentioned languages and are connected to the specified interfaces are allowed.

The following additional peripherals can be used with the TOE preserving the security functionality:

- all terminals supported by the TOE software (except hot pluggable devices connected via USB or IEEE 1394 (Firewire) interfaces).
- all storage devices and backup devices supported by the TOE software (hard disks, CDROM and DVDROM drives, streamer drives, floppy disk drives) (except hot pluggable devices connected via USB or IEEE 1394 (Firewire) interfaces).

Note: peripheral devices are part of the TOE environment.

Note: the peripherals are physical peripherals for all systems.

Note: Excluding hot pluggable devices connected via USB does not exclude all USB devices. USB keyboards and mice may be attached provided they are connected before booting the operating system.

3 TOE Security Environment

3.1 Introduction

The statement of TOE security environment describes the security aspects of the environment in which the TOE is intended to be used and the manner in which it is expected to be deployed.

To this end, the statement of TOE security environment identifies the list of assumptions made on the operational environment (including physical and procedural measures) and the intended method of use of the product, defines the threats that the product is designed to counter, and the organizational security policies with which the product is designed to comply.

3.2 Threats

The assumed security threats are listed below.

The **IT assets** to be protected comprise the information stored, processed or transmitted by the TOE. The term “information” is used here to refer to all data held within a server.

The TOE counters the general threat of unauthorized access to information, where "access" includes disclosure, modification and destruction.

The **threat agents** can be categorized as either:

- unauthorized users of the TOE, i.e. individuals who have not been granted the right to access the system; or
- authorized users of the TOE, i.e. individuals who have been granted the right to access the system.

The threat agents are assumed to originate from a well managed user community in a non-hostile working environment, and hence the product protects against threats of obvious security vulnerabilities that might be exploited in the intended environment for the TOE. The TOE in accordance with the strength of function claimed protects against straightforward or intentional breach of TOE security by attackers possessing a low attack potential.

The threats listed below are grouped according to whether or not they are countered by the TOE. Those that are not countered by the TOE are countered by environmental or external mechanisms.

3.2.1 Threats countered by the TOE

The following threats have been gathered from the different protection profiles this ST claims conformance to. Rather than combining similar threats into one, this ST lists the different threats as found in the respective PPs, recognizing that they are overlapping in several areas.

In all modes of operation, the TOE counters these threats:

T.UAUSER An attacker (possibly, but not necessarily, an unauthorized user of the TOE) may impersonate an authorized user of the TOE. This includes the threat of an authorized user that tries to impersonate as another authorized user without knowing the authentication information.

T.UAACCESS An authorized user of the TOE may access information resources without having permission from the person who owns, or is responsible for, the information resource for the type of access.

T.COMPROT An attacker (possibly, but not necessarily, an unauthorized user of the TOE) may intercept a communication link between the TOE and another trusted IT product to read or modify information transferred between the TOE and the other trusted IT product (which may be another instantiation of the TOE) using defined protocols (SSH or SSL) in a way that can not be detected by the TOE or the other trusted IT product.

In LSPP/RBAC mode, the TOE additionaly counters these threats:

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

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

3.2.2 Threats to be countered by measures within the TOE environment

The following threats to the system need to be countered in the TOE environment:

TE.HWMF An attacker with legitimate physical access to the hardware of the TOE (examples are maintenance personnel or legitimate users) or environmental conditions may cause a hardware malfunction with the effect that a user (normal or administrative) is losing stored data due to this hardware malfunction. An attacker may cause such a hardware malfunction either by having physical access to the hardware the TOE is running on or by executing software that capable of causing hardware malfunction. Note that such a hardware malfunction may be caused accidently without malicious intent by persons having physical access to the TOE.

TE.COR_FILE An attacker (including but not limited to an unauthorized user of the TOE) or environmental conditions such as a hardware malfunction may intentionally or accidentally modify or corrupt security enforcing or relevant files of the TOE without an administrative user being able to detect this. An attacker may corrupt such files either by having physical access to the TOE hardware, by booting other software than the TOE software in its evaluated configuration, or by modifying or corrupting files on backup media. Note that such a corruption may be caused accidently without malicious intent by persons having legitimate access to media where such data is stored.

3.3 Organizational Security Policies

The following organizational security policies (OSPs) have been gathered from the different protection profiles this ST claims conformance to. Rather than combining similar OSPs into one, this ST lists the different OSPs as found in the respective PPs, recognizing that they are overlapping in several areas.

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 organization must define a discretionary access control policy on a need-to-know basis which can be modeled based on:

a) the owner of the object; and

b) the identity of the subject attempting the access; and

c) the implicit and explicit access rights to the object granted to the subject by the object owner or an administrative user.

P.ACCOUNTABILITY

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

P.ACCESS (LSPP/RBAC mode only)

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

P.CLASSIFICATION (LSPP/RBAC mode only)

The system must limit the access to information based on sensitivity, as represented by a label, of the information contained in objects, and the formal clearance of users, as represented by subjects, to access that information. The access rules enforced prevent a subject from accessing information which is of higher sensitivity than it is operating at and prevent a subject from causing information from being downgraded to a lower sensitivity.

Note: The method for classification of information is made based on criteria set forth by the organization. This is usually done based on relative value to the organization and its interest in limiting dissemination of that information. The determination of classification of information is outside the scope of the IT system; the IT system is only expected to enforce the classification rules, not determine classification. The method for determining clearances is also outside

the scope of the IT system. It is essentially based on the trust placed in individual users by the organization. To some extent, it is also dependent upon the individual's role within the organization.

3.4 Assumptions

This section indicates the minimum physical and procedural measures required to maintain security of the Red Hat Enterprise Linux product.

3.4.1 Physical Aspects

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.

3.4.2 Personnel Aspects

A.MANAGE It is assumed that there are one or more competent individuals who are assigned to manage the TOE and the security of the information it contains.
In LSPP/RBAC mode, these individuals will have sole responsibility for the following functions:
(a) create and maintain roles
(b) establish and maintain relationships among roles
(c) Assignment and Revocation of users to roles.
In addition these individuals (as 'owners of the entire corporate data'), along with object owners will have the ability to assign and revoke object access rights to roles.

A.NO_EVIL_ADMIN 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.

A.UTRAIN Authorized users are trained to use the security functionality provided by the system appropriately.

A.UTRUST Authorized users are trusted to accomplish some task or group of tasks within a secure IT environment by exercising complete control over their data.

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

A.OWNER (LSPP/RBAC mode only): A limited set of users is given the rights to "create new data objects" and they become owners for those data objects. The organization is the owner of the rest of the information under the control of TOE.

3.4.3 Procedural Assumptions

A. CLEARANCE (LSPP/RBAC mode only): Procedures exist for granting users authorization for access to specific security levels.

A. SENSITIVITY (LSPP/RBAC mode only): Procedures exist for establishing the security level of all information imported into the system, for establishing the security level for all peripheral devices (e.g., printers, tape drives, disk drives) attached to the TOE, and marking a sensitivity label on all output generated.

3.4.4 Connectivity Aspects

A.NET_COMP All network components (such as bridges and routers) are assumed to correctly pass data without modification.

A.PEER	Any other system with which the TOE communicates is assumed to be under the same management control and operate under the same security policy constraints. There are no security requirements which address the need to trust external systems or the communications links to such systems.
A.CONNECT	All connections to peripheral devices and all network connections not using the secured protocols SSH v2 or SSL v3 reside within the controlled access facilities. Internal communication paths to access points such as terminals or other systems are assumed to be adequately protected.

4 Security Objectives

4.1 Security Objectives for the TOE

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

O.DISCRETIONARY_ACCESS The TSF must control access to resources based on identity of users. The TSF must allow authorized users to specify which resources may be accessed by which users.

O.MANDATORY_ACCESS (LSPP/RBAC mode only): The TSF must control access to resources based upon the sensitivity and categories of the information being accessed and the clearance of the subject attempting to access that information.

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_INFO The TOE 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 administrative users that are responsible for the management of TOE security and must ensure that only administrative users are able to access such functionality.

O.ENFORCEMENT The TSF must be designed and implemented in a manner which ensures that the organizational policies are enforced in the target environment The TOE security policy is enforced in a manner which ensures that the organizational policies are enforced in the target environment i.e. the integrity of the TSF is protected.

O.COMPROT The TSF must be designed and implemented in a manner that allows for establishing a trusted channel between the TOE and another trusted IT product that protects the user data transferred over this channel from disclosure and undetected modification.

O.DUTY (LSPP/RBAC mode only): The TOE must provide the capability of enforcing ‘separation of duties’, so that no single user has to be granted the right to perform all operations on important information.

O.HIERARCHICAL (LSPP/RBAC mode only): The TOE must allow hierarchical definitions of roles. Hierarchical definition of roles means the ability to define roles in terms of other roles. This saves time and allows for more convenient administration of the TOE.

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

4.2 Security Objectives for the TOE Environment

All security requirements listed in this section are targeted at the non-IT environment of the TOE.

OE.ADMIN Those responsible for the administration of the TOE are competent and trustworthy individuals, capable of managing the TOE and the security of the information it contains.

OE.CREDEN Those responsible for the TOE must ensure that user authentication data is stored securely and not disclosed to unauthorized individuals. In particular:

Procedures must be established to ensure that user passwords generated by an administrator during user account creation or modification are distributed in a secure manner, as appropriate for the purpose of the system.

The media on which authentication data is stored must not be physically removable from the system by other than administrative users.

Users must not disclose their passwords to other individuals.

OE.INSTALL Those responsible for the TOE must establish and implement procedures to ensure that the hardware, software and firmware components that comprise the system are distributed, installed and configured in a secure manner.

OE.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.

OE.INFO_PROTECT Those responsible for the TOE must establish and implement procedures to ensure that information is protected in an appropriate manner. In particular:

- DAC protections on security critical files (such as configuration files and authentication databases) shall always be set up correctly.
- Network and peripheral cabling must be approved for the transmittal of the most sensitive data held by the system. Such physical links are assumed to be adequately protected against threats to the confidentiality and integrity of the data transmitted unless one of the secure protocols provided by the TOE is used for the communication with another trusted entity.
- This requires that users are trained to perform those tasks properly and trustworthy to not deliberately misuse their access to information and pass it on to somebody that does not have the right to access the information.

OE.MAINTENANCE Administrative users of the TOE must ensure that any diagnostics facilities provided by the product are invoked at every scheduled preventative maintenance period.

OE.RECOVER Those responsible for the TOE must ensure that procedures and/or mechanisms are provided to assure that, after system failure or other discontinuity, recovery without a protection (i.e., security) compromise is obtained.

OE.SOFTWARE_IN Those responsible for the TOE shall ensure that the system shall be configured so that only an administrative user can introduce new trusted software into the system.

OE.SERIAL_LOGIN Those responsible for the TOE shall implement procedures to ensure that users clear the screen before logging off where serial login devices are used.

OE.CLASSIFICATION (LSPP mode only) Those responsible for the TOE must ensure that users of the TOE are cleared for access to information depending on the classification of the information. They must also ensure that information is correctly classified to be protected by the security functions of the TOE.

The following security objective applies in environments where specific threats to networked systems need to be countered. (Either physical protection measures or cryptographic controls may be applied to achieve this objective. The TOE provides some security functions that can be used to protect communication links, but the TOE does not enforce that those functions are used for all communication links. Communication links not protected by the functions provided as part of the TOE or communication links that need protection against interruption of communication have to be protected by security measures in the TOE environment.)

OE.PROTECT Those responsible for the TOE must ensure that procedures and/or mechanisms exist to ensure that data transferred between servers is secured from disclosure and tampering when using communication links not protected by the use of the SSL or SSH protocols. (Note that interruption of communication is not prevented by the use of those protocols. If protection against interruption of communication is required, adequate protection in the TOE environment has to be established for all communication links.).

5 Security Requirements

5.1 TOE Security Functional Requirements

Most of the following security functional requirements are taken from the "Controlled Access Protection Profile", [CAPP], the "Labeled Access Protection Profile" [LSPP] and the "Role-Based Accees Control Protection Profile" [RBACPP]. The requirement FMT_SMF.1 was added due to an added dependency in [CC]. The requirements FCS_CKM.1, FCS_CKM.2, FCS_COP.1, FDP_UCT.1, FDP_UIT.1, FMT_MSA.2 and FTP_ITC.1 represent TOE-specific extensions to the requirements defined by the protection profiles.

[CAPP], [LSPP] and [RBACPP] have already performed some instantiations and even some refinements of the security functional requirements as defined in the Common Criteria. Those instantiations and refinements are marked in **bold** within each of the requirements. In addition this Security Target has instantiated and refined the requirements as stated in [CAPP], [LSPP] and [RBACPP]. Those instantiations and refinements that are specific for this Security Target are marked in ***bold, italic and blue***.

Security functional requirements in addition to those taken from [CAPP], [LSPP] or [RBACPP] are shown in green with TOE-specific instantiations marked in ***bold, italic and green***.

5.1.1 Security Audit (FAU)

5.1.1.1 Audit Data Generation (FAU_GEN.1)

FAU_GEN.1.1 The TSF shall be able to generate an audit record of the auditable events **listed in column "Event" of *Table 5-1* (Auditable Events). This includes all auditable events for the basic level of audit, except FIA_UID.1's user identity during failures.**

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

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

b) The sensitivity labels of subjects, objects, or information involved;

c) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST the following information

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

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

d) The additional information specified in the "Details" column of *Table 5-1* (Auditable Events).

Table 5-1: Auditable Events

Component	Event	Details (Event Names)
FAU_GEN.1	Start-up and shutdown of the audit functions.	Events DAEMON_START, DAEMON_END, generated by auditd
FAU_GEN.2	None	
FAU_SAR.1	Reading of information from the audit records.	syscall *open* (on the audit log files)

Component	Event	Details (Event Names)
FAU_SAR.2	Unsuccessful attempts to read information from the audit records.	like FAU_SAR.1, but with negative results
FAU_SAR.3	None	
FAU_SEL.1	All modifications to the audit configuration that occur while the audit collection functions are operating.	Events DAEMON_CONFIG, DAEMON_RECONFIG generated by `auditd`; syscalls *open*, *link*, *unlink*, *rename*, *truncate* (write access to configuration files)
FAU_STG.1[1]	None	
FAU_STG.3	Actions taken due to exceeding of a threshold.	Event "log file is larger than max size" or "low on disk space" (generated by `auditd`); execution of administrator-specified alert action (such as file rotation, switch to single user mode, or system halt) based on the settings of the *space_left_action* and *admin_space_left_action* configuration parameters for auditd
FAU_STG.4	Actions taken due to the audit storage failure.	Event "no space left" or "error writing an event to disk" (generated by `auditd`); execution of administrator-specified alert action (such as switch to single user mode or system halt that terminates all programs capable of generating auditable events) based on the *disk_full_action* and *disk_error_action* configuration parameters for auditd
FCS_CKM.1	None	
FCS_CKM.2	None	
FCS_COP.1	None	
FDP_ACC.1(1)	None	
FDP_ACC.1(2) LSPP/RBAC mode	None	
FDP_ACF.1(1)	All requests to perform an operation on an object covered by the SFP.	File system objects: syscalls *chmod*, *chown*, *setxattr*, *removexattr*, *link*, *symlink*, *mknod*, *open*, *rename*, *truncate*, *unlink*, *rmdir*, *mount*, *umount* Message queue objects: syscalls *msgctl*, *msgget* Semaphore objects: syscalls *semget*, *semctl*, *semop*, *semtimedop* Shared memory objects: syscalls *shmget*, *shmctl*; Details include the identity of the object.

[1] The erroneous reference of CAPP/LSPP to FAU_STG.2 has been corrected in this ST.

Component	Event	Details (Event Names)
FDP_ACF.1(2) LSPP/RBAC mode	All requests to perform an operation on an object covered by the SFP.	File system objects: syscalls *chmod, chown, setxattr, removexattr, link, symlink, mknod, open, rename, truncate, unlink, rmdir, mount, umount* Message queue objects: syscalls *msgctl, msgget* Semaphore objects: syscalls *semget, semctl, semop, semtimedop* Shared memory objects: syscalls *shmget, shmctl*; Details include the identity of the object.
FDP_ETC.1 LSPP/RBAC mode	All attempts to export information.	Event: system calls connecting to external data stores: *mount, accept, connect, sendto, sendmsg, open, umount*
FDP_ETC.2 LSPP/RBAC mode	All attempts to export information. Overriding of human-readable output marking. (Additional)	Event: system calls connecting to external data stores: *mount, accept, connect, sendto, sendmsg, open, umount* In addition, applications create audit entries (like *star* or the printer subsystem)
FDP_IFC.1 LSPP/RBAC mode	None.	
FDP_IFF.2 LSPP/RBAC mode	All decisions on requests for information flow.	Event: error codes of system calls accessing objects under the SFP are audited and indicate the decisions on information flow (either a successful system calls indicates a positive decision, whereas *EACCESS* or *EPERM* indicates a permission denial)
FDP_ITC.1 LSPP/RBAC mode	All attempts to import user data, including any security attributes.	Event: system calls connecting to external data stores: *mount, accept, connect, sendto, sendmsg, open, umount*
FDP_ITC.2 LSPP/RBAC mode	All attempts to import user data, including any security attributes.	Event: system calls connecting to external data stores: *mount, accept, connect, sendto, sendmsg, open, umount* In addition, applications create audit entries (like *star*)
FDP_RIP.2	None	
Note 1	None	
FDP_UCT.1	None	
FDP_UIT.1	None	
FIA_ATD.1	None	

Component	Event	Details (Event Names)
FIA_SOS.1	Rejection or acceptance by the TSF of any tested secret.	Event USER_AUTH and USER_CHAUTHTOK (from PAM framework); details include origin of attempt (terminal or IP address as applicable)
FIA_UAU.2	All use of the authentication mechanism.	Event USER_AUTH and USER_CHAUTHTOK (from PAM framework)
FIA_UAU.7	None	
FIA_UID.2	All use of the user identification mechanism, including the identity provided during successful attempts.	Events USER_AUTH and USER_CHAUTHTOK(from PAM framework)
FIA_USB.1	Success and failure of binding user security attributes to a subject (e.g. success and failure to create a subject).	Event LOGIN (from PAM framework); syscalls fork, vfork and clone Failure: Events LOGIN (from PAM framework, failure status)
FMT_MSA.1(1)	All modifications of the values of security attributes.	syscalls chmod, chown, setxattr, msgctl, semctl, shmctl, removexattr
FMT_MSA.1(2) LSPP/RBAC mode	All modifications of the values of security attributes.	Event: audit message from the application *load_policy* and syscall *open* on selinuxfs files
FMT_MSA.1(3) LSPP/RBAC mode	All modifications of the values of security attributes.	Event: audit message from the application *load_policy* and syscall *open* on selinuxfs files
FMT_MSA.2	All offered and rejected values for a security attribute.	Events USER_AUTH and USER_CHAUTHTOK(from PAM framework) Failure: Events LOGIN (from PAM framework, failure status)
FMT_MSA.3(1)	Modifications of the default setting of permissive or restrictive rules. All modifications of the initial value of security attributes.	syscalls umask, open
FMT_MSA.3(2) LSPP/RBAC mode	Modifications of the default setting of permissive or restrictive rules. All modifications of the initial value of security attributes.	Event: audit message from the application *load_policy* and syscall *open* on selinuxfs files
FMT_MSA.3(3) LSPP/RBAC mode	Modifications of the default setting of permissive or restrictive rules. All modifications of the initial value of security attributes.	Event: audit message from the application *load_policy* and syscall *open* on selinuxfs files
FMT_MTD.1(1) Audit Trail	All modifications to the values of TSF data.	syscalls open, rename, link, unlink, truncate (of audit log files)
FMT_MTD.1(2) Audit Events	All modifications to the values of TSF data.	syscalls open, link, rename, truncate, unlink (of audit config files); event ”config change”
FMT_MTD.1(3) User Attributes	All modifications to the values of TSF data.	audit text messages from “shadow-utils” trusted programs, details include new value of of the TSF data

Component	Event	Details (Event Names)
FMT_MTD.1(4) Authentication Data	All modifications to the values of TSF data.	audit text messages from "shadow-utils" trusted programs including USER_CHAUTHTOK PAM messages; attempts to bypass trusted programs detected through audited syscalls *open*, *rename*, *truncate*, *unlink*
FMT_MTD.1(5) LSPP/RBAC mode	All modifications to the values of TSF data, including: (i) Assignment of Users, Roles and Privileges to Roles (ii) Deletion of Users, Roles and Privileges from Roles (iii) Creation and Deletion of Roles	Event: audit message from the application *load_policy* and syscall *open* on selinuxfs files
FMT_MTD.3 LSPP/RBAC mode	All rejected values of TSF data.	Failure: Events LOGIN (from PAM framework, failure status)
FMT_REV.1(1)	All attempts to revoke security attributes.	Event: audit text messages from "shadow-utils" trusted programs; attempts to bypass trusted programs detected through audited syscalls *open*, *rename*, *truncate*, *unlink*
FMT_REV.1(2)	All modifications to the values of TSF data.	system calls *chmod*, *chown*, *setxattr*, *removexattr*, *unlink*, *truncate*, *msgctl*, *semctl*, *shmctl*
FMT_SMR.2	Modifications to the group of users that are part of a role.	Event: audit text messages from "shadow-utils" trusted programs "group member added", "group member removed", "group administrators set", "group members set" (from trusted programs in shadow suite).
FMT_SMR.2	Every use of the rights of a role. (Additional / Detailed)	The user's actions result in audited syscalls and the use of trusted programs that are audited. Details include the login ID, the origin can be determined from the associated LOGIN record for this login ID and audit session ID.
FPT_AMT.1	Execution of the tests of the underlying machine and the results of the test.	Event messages "amtu-*" (generated by AMTU testing tool)
FPT_FLS.1 LSPP/RBAC mode	Failure of the TSF.	Failure: audit entries from applications managing the SELinux configuration and SELinux aware applications using libselinux: *crond*, *login*, *sshd*, *su*
FPT_RCV.1 LSPP/RBAC mode	Type of failure or service discontinuity.	Event/Failure: audit entry because of switch between multi-user mode and single-user mode
FPT_RCV.4 LSPP/RBAC mode	If possible, the detection of a failure of a security function.	Event/Failure: audit entry because of switch between multi-user mode and single-user mode
FPT_RVM.1	None	
FPT_SEP.1	None	
FPT_STM.1	Changes to the time.	Event: syscalls *settimeofday*, *adjtimex*, *clock_settime*. Also *USYS_CONFIG* messages from *hwclock*

Component	Event	Details (Event Names)
FPT_TDC.1 LSPP/RBAC mode	None	
FPT_TST.1 LSPP/RBAC mode	Execution of the TSF self tests and the results of the tests.	Event: audit message from the test application
FTA_LSA.1 LSPP/RBAC mode	All attempts at selecting a session security attributes;	Event: audit entry from an application linked against libselinux: *crond, login, sshd, su*
FTA_TSE.1 LSPP/RBAC mode	All attempts at establishment of a user session.	Event: audit entry from an application linked against libselinux: *crond, login, sshd, su*
FTP_ITC.1	All attempted uses of the trusted channel functions. Identification of the initiator and target of all trusted channel functions.	Event: syscall *exec* (of `stunnel` program)

Application Note: The table lists the names of the events associated with the SFR. Details of the event specific data recorded with each event are defined in the audit design documentation.

5.1.1.2 User Identity Association (FAU_GEN.2)

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

Application Note: The TOE maintains a "Login ID", which is inherited by every new process spawned. This allows the TOE to identify the "real" originator of an event, regardless if he has changed his real and / or effective and filesystem user ID e. g. using the su command or executing a SUID or SGID program.

5.1.1.3 Audit Review (FAU_SAR.1)

FAU_SAR.1.1 The TSF shall provide **authorized administrators** with the capability to read **all audit information** from the audit records, ***including***:

a) Date and Time of Audit Event

b) The UserID responsible for the Event and optionally the role membership which enabled the user to perform the event successfully

c) The access control operation and the object on which it was performed.

d) The outcome of the event (success or failure)

e) The User Session Identifier or Terminal Type

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

5.1.1.4 Restricted Audit Review (FAU_SAR.2)

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

Application Note: DAC controls ensure that only administrative users have access to the audit records.

5.1.1.5 Selectable Audit Review (FAU_SAR.3)

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

a) User identity;

b) (LSPP/RBAC mode) Subject sensitivity label;

c) (LSPP/RBAC mode) Object sensitivity label;

d) Date and Time of Audit event

e) Object Name & type of access

f) (LSPP/RBAC mode) Role that enabled the access

g) (LSPP/RBAC mode) Any combination of the above items (a), (d), (e) or (f).

h) group identifier (real and effective)

i) event type

j) outcome (success/failure)

k) login from specific remote hostname

l) audit id

m) process id

5.1.1.6 Selective Audit (FAU_SEL.1)

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) Subject sensitivity label;

c) Object sensitivity label;

d) Object identity, user identity, subject identity, host identity, and event type

e) Users belonging to a specified Role and Access types (e.g. delete, insert) on a particular object

f) system call number

g) directory or file name.

Application Note: The TOE provides the administrator the ability to select the events to audit. This can be done by the administrator editing the filter configuration file of the audit daemon and then using the */etc/rc.d/init.d/auditd* script with the 'reload' parameter to notify the audit daemon of the change in the configuration. The audit daemon in turn notifies the kernel of the new auditing policy.

5.1.1.7 Guarantees of Audit Data Availability (FAU_STG.1)

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

FAU_STG.1.2 The TSF shall be able to **prevent** modifications to the audit records.

Application Note: This is achieved using the DAC controls.

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

FAU_STG.3.1 The TSF shall **generate an alarm to the authorized administrator** if the audit trail exceeds ***a value defined in the file /etc/auditd.conf for the minimum space required for the file system the audit log file resides in***.

Application Note: The alarm generated by the TOE is a syslog message. This message is generated when the audit trail capacity exceeds the limit defined in the auditd.conf file. This limit can be defined by the system administrator by editing the auditd.conf file and then reloading the audit configuration.

5.1.1.9 Prevention of Audit Data Loss (FAU_STG.4)

FAU_STG.4.1 The TSF shall **be able to prevent auditable events, except those taken by the authorized administrator**, and ***no other actions,*** if the audit trail is full.

Application Note: If the audit trail gets full, the audit daemon will execute an administrator-defined action. The possible actions include a switch to single user mode or system halt, each of these will terminate all processes capable of generating auditable events. The system administrator can then back up the audit trail and make space available for the audit trail, then restart the TOE in multi-user mode.

5.1.2 Cryptographic Support (FCS)

5.1.2.1 Cryptographic key generation (SSL: Symmetric algorithms) (FCS_CKM.1(1))

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm ***as defined in the SSL v3 standard*** and specified cryptographic key sizes ***128 bit (RC4) , 168 bit (TDES), 128 bit (AES), 256 bit (AES)*** that meet the following: ***generation and exchange of session keys as defined in the SSL v3 and standard with the cipher suites defined in FCS_COP.1(2).***

Application Note: Generation of symmetric keys is defined in section 6.2 in the SSL v3 standard. The OpenSSL library used by the TOE also supports SSL v2, but this is seen as being not part of the evaluated configuration. The evaluation will assess that the keys are generated in accordance with the requirements defined in the SSL v3 standard. With respect to the strength of function, no assessment of the strength of the cryptographic algorithm itself and no analysis for potential weaknesses of keys with respect to the algorithm are performed. The key generation process will only be analysed and rated with respect to the entropy of the input to the key generation process and with respect to the fact that any postprocessing of this input will maintain the entropy

5.1.2.2 Cryptographic key generation (SSH: Symmetric algorithms) (FCS_CKM.1(2))

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm ***as defined in the SSH v2 standard SSH Transport Layer Protocol [SSH-TRANS]*** and specified cryptographic key sizes ***168 bit (TDES)*** that meet the following: ***generation and exchange of session keys as defined in the [SSH-TRANS] standard using the Diffie-Hellman key negotiation protocol.***

Application Note: For details of the key generation / key negotiation process see section 4.5, chapters 6.5, 6.6, and 7 of the SSH Transport Layer Protocol specification [SSH-TRANS] as published by the Secure Shell Charter of the Internet Engineering Task Force (IETF). The evaluation will assess that the keys are generated in accordance with the requirements defined in the [SSH-TRANS] standard, but no assessment on the strength of the keys generated will be performed as part of this evaluation.

5.1.2.3 Cryptographic key generation (SSL: RSA) (FCS_CKM.1(3))

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm ***product specific*** and specified cryptographic key sizes ***1024 bit*** that meet the following: ***not specified***

Application Note: The SSL v3 specification does not define how the RSA key pair is generated. This is up to the implementation. Almost all implementations of the SSL v3 standard have their own algorithm for RSA key pair generation (if they support cipher suites that use RSA). Therefore the key generation and algorithm and the standard to follow are not defined here. Only the required key size is specified. The evaluation will assess that the keys generated form a correct RSA key pair. No assessment on the strength of the keys generated will be performed as part of this evaluation. The only assessment made is with respect to the probability of the numbers used to be prime.

5.1.2.4 Cryptographic key distribution (SSL: RSA public keys) (FCS_CKM.2(1))

FCS_CKM.2.1 The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method ***digital certificates for public RSA keys*** that meets the following: ***certificate format as defined in the standard X.509 Version 3***.

Application Note: This requirement addresses the exchange of public RSA keys as part of the SSL client and server authentication.

5.1.2.5 Cryptographic key distribution (SSH: Diffie-Hellman key negotiation) (FCS_CKM.2(2))

FCS_CKM.2.1 The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method ***diffie-hellman-group1-sha1*** that meets the following: ***Specification in [SSH-TRANS]***.

Application Note: The Diffie-Hellman protocol can be seen as a combined way to generate and distribute a shared session key between two communicating parties. So the Diffie-Hellman algorithm used by SSH is mentioned both in the key generation as well as in the key distribution security functional requirement.

5.1.2.6 Cryptographic key distribution (SSH: DSS public keys) (FCS_CKM.2(3))

FCS_CKM.2.1 The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method ***digital certificates for public DSS keys*** that meets the following: ***ssh-dss key format as defined in [SSH-TRANS]***.

5.1.2.7 Cryptographic key distribution (SSL: Symmetric keys) (FCS_CKM.2(4))

FCS_CKM.2.1 The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method ***Secure Socket Layer handshake using RSA encrypted exchange of session keys*** that meets the following: ***SSL Version 3 [SSLv3]***.

Application Note: This requirement addresses the exchange of SSL session keys as part of the SSL handshake protocol.

5.1.2.8 Cryptographic operation (RSA) (FCS_COP.1(1))

FCS_COP.1.1 The TSF shall perform ***digital signature generation and digital signature verification*** in accordance with a specified cryptographic algorithm ***RSA*** and cryptographic key sizes ***1024 bit*** that meet the following: ***SSL Version 3[SSLv3].***

Application Note: This requirement addresses the RSA digital signature generation and verification operations using the RSA algorithm as required by the SSL session establishment protocol (provided a cipher suite including RSA is used). Note that the details of the signature format such as the use of the PKCS#1 block type 1 and block type 2 are defined in the SSL Version 3 standard.

5.1.2.9 Cryptographic operation (SSL: Symmetric operations) (FCS_COP.1(2))

FCS_COP.1.1 The TSF shall perform ***encryption and decryption*** in accordance with a specified cryptographic algorithm ***RC4, TDES and AES*** and cryptographic key sizes ***128 bit (RC4), 168 bit (TDES), 128 bit (AES) and 256 bit (AES)*** that meet the following: ***SSL Version 3 [SSLv3] and the following cipher suites: SSL_RSA_WITH_RC4_128_SHA, SSL_RSA_WITH_3DES_EDE_CBC_SHA, as defined in the SSL v3 standard, and TLS_RSA_WITH_AES_128_CBC_SHA, TLS_RSA_WITH_AES_256_CBC_SHA as defined in the IETF RFC 3268.***

5.1.2.10 Cryptographic operation (SSH: Symmetric operations) (FCS_COP.1(3))

FCS_COP.1.1 The TSF shall perform ***encryption and decryption*** in accordance with a specified cryptographic algorithm ***TDES*** and cryptographic key sizes ***168 bit (TDES)*** that meet the following: ***SSH Version 2 [SSH-TRANS] and the following cipher suite: 3des-cbc as defined in [SSH-TANS].***

5.1.3 User Data Protection (FDP)

5.1.3.1 Discretionary Access Control Policy (FDP_ACC.1(1))

FDP_ACC.1.1 The TSF shall enforce the **Discretionary Access Control Policy** on ***processes*** acting on the behalf of users ***as subjects and file system objects (ordinary files, directories, symbolic links, device special files, UNIX Domain socket special files, named pipes), IPC objects (message queues, semaphores, shared memory segments)*** **and all operations among subjects and objects covered by the DAC policy.**

5.1.3.2 Role-based Access Control Policy (FDP_ACC.1(2)) (LSPP/RBAC mode only)

FDP_ACC.1.1 The TSF shall enforce the **Role-based Access Control (RBAC) SFP** on:

a) Subjects covered by RBAC SFP: ***processes***

b) Objects covered by RBAC SFP: ***file system objects and IPC objects***

c) All Operations on Objects covered by RBAC SFP

5.1.3.3 Discretionary Access Control Functions (FDP_ACF.1(1))

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

a) The ***filesystem*** **user identity and group membership(s) associated with a subject; and**

b) The following access control attributes associated with an object:

File system objects:

POSIX ACLs and permission bits.
(ACLs can be used to grant or deny access to the granularity of a single user or

group using Access Control Entries. Those ACL entries include the standard Unix permission bits. Posix ACLs can be used for file system objects within the ext3 file system).

Access rights for file system objects are:

- read
- write
- execute (ordinary files)
- search (directories)

IPC objects:

permission bits

Access rights for IPC objects are:

- read
- write

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

File system objects within the ext3 file system:

A subject must have search permission for every element of the pathname and the requested access for the object. A subject has a specific type access to an object if:

- ***The subject has been granted access according to the ACL_USER_OBJ or ACL_OTHER type entry in the ACL of the object***

Or

- ***The subject has been granted access by an ACL_USER, ACL_GROUP_OBJ or ACL_GROUP entry and the associated right is also granted by the ACL_MASK entry of the ACL if the ACL_MASK entry exist***

Or

- ***The subject has been granted access by the ACL_GROUP_OBJ entry and no ACL_MASK entry exists in the ACL of the object.***

File system objects in other file systems:

A subject must have search permission for every element of the pathname and the requested access for the object. A subject has a specific type access to an object if:

- ***The subject has the filesystem userid of the owner of the object and the requested type of access is within the permission bits defined for the owner***

Or

- ***The subject has not the filesystem userid of the owner of the object but the filesystem group id identical to the file system objects group id and the requested type of access is within the permission bits defined for the group***

Or

- ***The subject has neither the filesystem userid of the owner of the object nor is the filesystem group id identical to the file system object group id and requested type of access is within the permission bits defined for "world"***

IPC objects:

Access permissions are defined by permission bits of the IPC object. The process creating the object defines the creator, owner and group based on the userid of the current process. Access of a process to an IPC object is allowed, if

- ***the effective userid of the of the current process is equal to the userid of the IPC object creator or owner and the „owner" permission bit for the requested type of access is set or***
- ***the effective userid of the current process is not equal to the userid of the IPC object creator or owner and the effective group id of the current process is equal to the group id of the IPC object and the „group" permission bit for the requested type of access is set or***
- ***The „world" permission bit for the requested type of access is set for users that do not satisfy one of the first two conditions***

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

File System Objects:

A process with a user ID of 0 is known as a root user process. These processes are generally allowed all access permissions. But if a root user process requests execute permission for a program (as a file system object), access is granted only if execute permission is granted to at least one user.

IPC objects:

A process with a user ID of 0 is known as a root user process. These processes are generally allowed all access permissions.

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the ***following rules:***

Write access to file system objects other than device special files on a file system mounted as read-only is always denied.
Write access to a file marked as immutable is always denied.

Application note: The TOE includes functionality that can add additional restrictions, this ST does not make any claims about these additional checks.

5.1.3.4 Role-based Access Control Functions (FDP_ACF.1(2)) (LSPP/RBAC mode only)

FDP_ACF.1.1 The TSF shall enforce the **RBAC SFP** to objects based on the following **user attributes:**

a) User Identity

b) Authorized Roles for the User (refer to Glossary for definition)

The TSF shall enforce the **RBAC SFP** to objects based on the following **subject attributes:**

a) Subject Identity

b) Role(s) which can invoke the subject

The TSF shall enforce the **RBAC SFP** to objects based on the following **object attributes:**

a) Object Identity

b) Operations permitted on the objects for various Roles

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

FDP_ACF.1.3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: **Allow an access operation by a subject on an object only if the user associated with the subject belongs to a role that permits the access operation on the object.**

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the **user associated with the subject not belonging to any role that permits the requested access operation on the object.**

5.1.3.5 Export of unlabeled user data (FDP_ETC.1) (LSPP/RBAC mode only)

FDP_ETC.1.1 The TSF shall enforce the **Mandatory Access Control Policy** when exporting ***unlabeled*** user data, controlled under the **MAC policy**, outside the TSF Scope of Control (TSC).

FDP_ETC.1.2 The TSF shall export the ***unlabeled*** user data without the user data's associated security attributes.

Note6 **The TSF shall enforce the following rules when unlabeled user data is exported from the TSC:**

a) devices used to export data without security attributes cannot be used to export data with security attributes unless the change in device state is performed manually and is auditable;

b) ***none.***

5.1.3.6 Export of labeled user data (FDP_ETC.2) (LSPP/RBAC mode only)

FDP_ETC.2.1 The TSF shall enforce the **Mandatory Access Control Policy** when exporting ***labeled*** user data, controlled under the **MAC policy**, outside the TSC.

FDP_ETC.2.2 The TSF shall export the ***labeled*** user data with the user data's associated security attributes.

FDP_ETC.2.3 The TSF shall ensure that the security attributes, when exported outside the TSC, are unambiguously associated with the exported ***labeled*** user data.

FDP_ETC.2.4 The TSF shall enforce the following rules when ***labeled*** user data is exported from the TSC:

a) when data is exported in a human-readable or printable form:

- **the authorized administrator shall be able to specify the printable label that is assigned to the sensitivity label associated with the data.**
- **each print job shall be marked at the beginning and end with the printable label assigned to the "least upper bound" sensitivity label of all the data exported in the print job.**

- **each page of printed output shall be marked with the printable label assigned to the "least upper bound" sensitivity label of all the data exported to the page. By default, this marking shall appear on both the top and bottom of each printed page.**

b) devices used to export data with security attributes cannot be used to export data without security attributes unless the change in device state is performed manually and is auditable;

c) devices used to export data with security attributes shall completely and unambiguously associate the security attributes with the corresponding data;

d) *none*.

Application note: Labeled data can be exported using the star utility, which will keep the labels attached to file system objects, as it preserves the extended file attributes

5.1.3.7 Mandatory access control policy (FDP_IFC.1) (LSPP/RBAC mode only)

FDP_IFC.1.1 The TSF shall enforce the **Mandatory Access Control Policy** on ***processes acting on behalf of users as subjects, file system objects and IPC objects as objects*, and all operations among subjects and objects covered by the MAC policy.**

5.1.3.8 Mandatory access control functions (FDP_IFF.2) (LSPP/RBAC mode only)

FDP_IFF.2.1 The TSF shall enforce the **Mandatory Access Control Policy** based on the following types of subject and information security attributes:

a) the sensitivity label of the subject; and

b) the sensitivity label of the object containing the information.

Sensitivity label of subjects and objects shall consist of the following:

- **a hierarchical level; and**
- **a set of non-hierarchical categories.**

FDP_IFF.2.2 The TSF shall permit an information flow between a controlled subject and controlled information through a controlled operation if the following rules, based on the ordering relationships between security attributes, hold:

a) if the sensitivity label of the subject is greater than or equal to the sensitivity label of the object, the flow of information from the object to the subject is permitted (a read operation);

b) if the sensitivity label of the object is greater than or equal to the sensitivity label of the subject; the flow of information from the subject to the object is permitted (a write operation);

c) if the sensitivity label of subject A is greater than or equal to the sensitivity label of subject B; the flow of information from subject B to subject A is permitted.

FDP_IFF.2.3 The TSF shall enforce the: ***none***

FDP_IFF.2.4 The TSF shall provide the following: ***none***

FDP_IFF.2.5 The TSF shall explicitly authorize an information flow based on the following rules: a user is permitted to bypass the information flow policy, ***if the subject holds a MAC override privilege for the requested operation and object type***.

Application note: All MAC override privileges are assigned to the security administrator role and to some restore and installation utilities like *rpm*.

FDP_IFF.2.6 The TSF shall explicitly deny an information flow based on the following rules: ***none***

FDP_IFF.2.7 The TSF shall enforce the following relationships for any two valid sensitivity labels:

a) there exists an ordering function that, given two valid sensitivity labels, determines if the sensitivity labels are equal, if one sensitivity label is greater than the other, or if the sensitivity labels are incomparable; and

- **sensitivity labels are equal if the hierarchical level of both labels are equal and the non-hierarchically category sets are equal.**
- **sensitivity label A is greater than sensitivity label B if one of the following conditions exists:**
  - **if the hierarchical level of A is greater than the hierarchical level of B, and the non-hierarchical category set of A is equal to the non-hierarchical category set of B.**
  - **if the hierarchical level of A is equal to the hierarchical level of B, and the non-hierarchical category set of A is a proper super-set of the nonhierarchical category set of B.**
  - **if the hierarchical level of A is greater than the hierarchical level of B, and the non-hierarchical category set of A is a proper[2] superset of the nonhierarchical category set of B.**
- **sensitivity labels are incomparable if they are not equal and neither label is greater than the other.**

b) there exists a "least upper bound" in the set of sensitivity labels, such that, given any two valid sensitivity labels, there is a valid sensitivity label that is greater than or equal to the two valid sensitivity labels; and

c) there exists a "greatest lower bound" in the set of the sensitivity labels, such that, given any two valid sensitivity labels, there is a valid sensitivity label that is not greater than the two valid sensitivity labels.

5.1.3.9 Import of unlabeled user data (FDP_ITC.1) (LSPP/RBAC mode only)

FDP_ITC.1.1 The TSF shall enforce the **Mandatory Access Control Policy** when importing ***unlabeled*** user data, controlled under the **MAC policy**, from outside the TSC.

FDP_ITC.1.2 The TSF shall ignore any security attributes associated with the ***unlabeled*** user data when imported from outside the TSC.

FDP_ITC.1.3 The TSF shall enforce the following rules when importing ***unlabeled*** user data controlled under the MAC policy from outside the TSC:

a) devices used to import data without security attributes cannot be used to import data with security attributes unless the change in device state is performed manually and is auditable.

b) *none.*

Application note: See the application note on FDP_ETC.1 for export of unlabeled data.

[2] The word "proper" in this rule has been taken from LSPP, but is definitively wrong in this rule. Because the hierarchical level of A is already greater than the hierarchical level of B, A is greater than B even if the sets of categories of A and B are identical

The requirement also applies for the import of RSA key pairs or Diffie-Hellman key pairs imported to be used for the cryptographic operations of the TOE. The administrators need to ensure using the MAC and DAC policy enforced by the TOE that this key material is imported in a secure way and can not be imported by unauthorized users.

5.1.3.10 Import of labeled user data (FDP_ITC.2) (LSPP mode only)

FDP_ITC.2.1 The TSF shall enforce the **Mandatory Access Control Policy** when importing ***labeled*** user data, controlled under the **MAC policy**, from outside the TSC.

FDP_ITC.2.2 The TSF shall use the security attributes associated with the imported ***labeled*** user data.

FDP_ITC.2.3 The TSF shall ensure that the protocol used provides for the unambiguous association between security attributes and the ***labeled*** user data received.

FDP_ITC.2.4 The TSF shall ensure that interpretation of the security attributes of the imported ***labeled*** user data is as intended by the source of the user data.

FDP_ITC.2.5 The TSF shall enforce the following rules when importing ***labeled*** user data controlled under the **MAC policy** from outside the TSC:

a) devices used to import data with security attributes cannot be used to import data without security attributes unless the change in device state is performed manually and is auditable;

b) ***none.***

c) sensitivity label, consisting of the following:

- **a hierarchical level; and**
- **a set of non-hierarchical categories.**

Application note: See the application note on FDP_ETC.2 for export of labeled data.

5.1.3.11 Object Residual Information Protection (FDP_RIP.2)

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

5.1.3.12 Subject Residual Information Protection (Note 1)

NOTE 1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the allocation of the resource to all subjects.

5.1.3.13 Basic data exchange confidentiality (FDP_UCT.1)

FDP_UCT.1.1 The TSF shall enforce the ***Discretionary Access Control Policy and the Mandatory Access Control Policy*** to be able to ***transmit and receive*** objects in a manner protected from unauthorised disclosure.

Application Note: Confidentiality of data during transmission is ensured when the one of the secured protocols ssh, or ssl are used. User processes are still bound by the discretionary access control policy with respect to the data they are able to transfer. The TOE is able act both as a server and a client for ssh, ssl, or IPSec connections.

5.1.3.14 Data exchange integrity (FDP_UIT.1)

FDP_UIT.1.1 The TSF shall enforce the ***Discretionary Access Control Policy Policy and the Mandatory Access Control Policy*** to be able to ***transmit and receive*** user data in a manner protected from ***modification and insertion*** errors.

FDP_UIT.1.2 The TSF shall be able to determine on receipt of user data, whether ***modification or insertion*** has occurred.

Application Note: Integrity of data during transmission is ensured when the one of the secured protocols ssh, ssl, or IPSec are used. User processes are still bound by the discretionary access control policy with respect to the data they are able to transfer. The TOE is able act both as a server and a client for ssh, ssl, or IPSec connections.

5.1.4 Identification and Authentication (FIA)

5.1.4.1 User Attribute Definition (FIA_ATD.1)

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

a) User Identifier;

b) Group Memberships;

c) Authentication Data;

d) *(LSPP/RBAC mode)* User Clearance;

e) *(LSPP/RBAC mode)* List of Authorized Roles;

f) (LSPP/RBAC mode) SELinux user identity;

g) (LSPP/RBAC mode) Default login security context.

5.1.4.2 Strength of Authentication Data (FIA_SOS.1)

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

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

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

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

5.1.4.3 Authentication (FIA_UAU.2)

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

Application Note: Untrusted processes running on behalf of a normal user may use network functions to import and export data they have access to. This process may therefore export user data without authenticating or even knowing the identity of a user receiving such data. This is not considered to be a violation of the security policy with respect to identification and authentication and discretionary access control, since it is well-known that discretionary access control can not control flow of information. An example of such an export function is a user process running a

web-server on an unprivileged port. Still this process is limited in its access by the security policy of the TOE.

5.1.4.4 Protected Authentication Feedback (FIA_UAU.7)

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

5.1.4.5 Identification (FIA_UID.2)

FIA_UID.2.1 The TSF shall require each user to identify itself before allowing any other TSF-mediated actions on behalf of that user.

5.1.4.6 User-Subject Binding (FIA_USB.1)

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

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

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

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

d) *(LSPP/RBAC mode)* The sensitivity label used to enforce the Mandatory Access Control Policy, which consists of the following:

- **A hierarchical level; and**
- **A set of non-hierarchical categories.**

e) (LSPP/RBAC mode) The Active Role set used to enforce the Role-based Access Control Policy as a subset of the Authorized Roles for the User.

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) *Upon successful identification and authentication, the login user ID, the real user ID, the filesystem user ID, and the effective user ID shall be those specified in the user entry for the user that has authenticated successfully.*

b) *Upon successful identification and authentication, the real group ID, the filesystem group ID, and the effective group ID shall be those specified via the group membership attribute in the user entry.*

c) *(LSPP/RBAC mode)* The sensitivity label associated with a subject shall be within the clearance range of the user.

d) *(LSPP/RBAC mode) The Active Role Set chosen during a login operation must be a subset of the Authorized Roles for the User.*

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) *The effective and filesystem user ID of a user can be changed by the use of an executable with the setuid bit set. In this case the program is executed with the effective and filesystem user ID of the program owner. Access rights are then evaluated using the filesystem user ID of the program owner. The real and login user ID remain unchanged.*

b) ***The effective, filesystem, and real user ID of a user can be changed by the su command. In this case the real, filesystem, and effective user ID of the user is changed to the user specified in the su command (provided authentication is successful). The login user ID remains unchanged.***

c) ***The filesystem and effective group ID of a user can be changed by the use of an executable with the setgid bit set. In this case the program is executed with the filesystem and effective group ID of the program owner. Access rights are then evaluated using the filesystem group ID of the program owner.***

d) ***(LSPP/RBAC mode) The active role of a subject can be changed if a transition rule for the requested change is defined and a allow rule for the requested transition is defined.***

e) ***(LSPP/RBAC mode) The sensitivity label of a subject can be changed if a transition rule for the requested change is defined and a allow rule for the requested transition is defined.***

5.1.5 Security Management (FMT)

5.1.5.1 Management of Object Security Attributes (FMT_MSA.1(1))

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 ***authorized administrators and the owner of the object. For IPC objects also the original creator of the object has the ability to modify the access control attributes.***

Application Note (LSPP/RBAC mode): authorized administrators are administrative users in the sysadm_r role

5.1.5.2 Management of object security attributes for MAC (FMT_MSA.1(2)) (LSPP/RBAC mode only)

FMT_MSA.1.1 The TSF shall enforce the **Mandatory Access Control Policy** to restrict the ability to modify the **sensitivity label associated with an object** to ***the security administrator***.

5.1.5.3 Management of User Security Attributes (FMT_MSA.1(3)) (LSPP/RBAC mode only)

FMT_MSA.1.1 The TSF shall enforce the **RBAC SFP** to restrict the ability to **modify, delete, create instances** of the **following user** security attribute to **a set of RBAC Administrative Roles**:

a) User Role Authorizations

FMT_MSA.1.1 The TSF shall enforce the **RBAC SFP** to restrict the ability to **create, modify the composition** of the **following user** security attribute to **a set of RBAC Administrative Roles**:

a) Default Active Role Set (refer to Glossary for definition)

FMT_MSA.1.1 The TSF shall enforce the **RBAC SFP** to restrict the ability to **modify the composition** of the **following session** security attribute to **session owner**:

a) Active Role set for a user (refer to Glossary for definition)

FMT_MSA.1.1 The TSF shall enforce the **RBAC SFP** to restrict the ability to **modify** the **object** security attributes to **(i) Object Owners and (ii) Set of RBAC Administrative Roles**.

5.1.5.4 Secure security attributes (FMT_MSA.2)

FMT_MSA.2.1 The TSF shall ensure that only secure values are accepted for security attributes.

Application Note: This SFR addresses several issues:

It fulfils a dependency for FCS_CKM.1, FCS_CKM.2, and FCS_COP.1. The assessment with respect to this requirement in the evaluation of this TOE does not include any assessment of the cryptographic strength of the keys generated or used. Instead the assessment with respect to this requirement just includes an assessment that the TOE protects those keys from unauthorized access, disclosure or tampering.

LSPP/RBAC mode only: It is also required by [RBACPP]; as such it addresses the ability of the system to enforce sufficiently strong passwords by applying password complexity rules to new passwords, and to reject illegal values for certain security attributes.

5.1.5.5 Static Attribute Initialization for DAC (FMT_MSA.3(1))

FMT_MSA.3.1 The TSF shall enforce the **Discretionary Access Control Policy** to provide **restrictive** default values for security attributes that are used to enforce the SFP. [3]

FMT_MSA.3.2 The TSF shall allow ***an authorized administrators and the owner of the object*** to specify alternative initial values to override the default values when an object or information is created.

Application note: Because the option to assign a property other than "restrictive" or "permissive" was only introduced with final interpretation RI#202, the authors of LSPP and CAPP have selected "restrictive", but allowed an authorized administrator to override those default values. In reality, most systems will neither define the "restrictive" nor the "permissive" case as the default value, but the default values will be defined such that they match the intended operational policy in the best way. This also applies to 5.1.5.6.

5.1.5.6 Static Attribute Initialization for MAC (FMT_MSA.3(2)) (LSPP/RBAC mode only)

FMT_MSA.3.1 The TSF shall enforce the **Mandatory Access Control Policy** to provide **restrictive** default values for security attributes that are used to enforce the SFP.[3]

FMT_MSA.3.2 The TSF shall allow the ***authorized administrator*** to specify alternative initial values to override the default values when an object or information is created.

5.1.5.7 Static Attribute Initialization for RBAC (FMT_MSA.3(3)) (LSPP/RBAC mode only)

FMT_MSA.3.1 The TSF shall enforce the **RBAC SFP** to provide ***the following*** default values for object security attributes that are used to enforce the SFP:

a) ***At creation time, an object is assigned a security context (SELinux user identity, role, type, sensitivity label) as defined in the transformation rules of the currently loaded policy.***

FMT_MSA.3.2 The TSF shall allow the **following** roles to specify alternative initial values to override the default values when an object or information is created:

a) Set of RBAC Administrative Roles

[3] [CAPP] and [LSPP] refined the term „SFP" to „Discretionary Access Control SFP" and „Mandatory Access Control SFP", respectively, interpreting a formatting error in previous CC versions as a requirement to instantiate the term SFP here. Since the formatting has been corrected in [CC], this instantiation has been removed in favor of more strict CC compliance. However, the title and wording of the SFRs clearly express that the restrictive default values only apply to the DAC and MAC SFP, respectively.

5.1.5.8 Management of the Audit Trail (FMT_MTD.1(1))

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

Application Note: This requirement is implemented using the discretionary access control features of the TOE to protect the files holding the audit trail.

5.1.5.9 Management of Audited Events (FMT_MTD.1(2))

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

Application Note: This requirement is implemented using the discretionary access control features of the TOE to protect the audit configuration files. In LSPP/RBAC mode, the authorized administrator is a user in the security administrator role.

5.1.5.10 Management of User Attributes (FMT_MTD.1(3))

FMT_MTD.1.1 The TSF shall restrict the ability to **initialize and modify** the **user security attributes, other than authentication data**, to **authorized administrators**.

5.1.5.11 Management of Authentication Data (FMT_MTD.1(4))

FMT_MTD.1.1 The TSF shall restrict the ability to **initialize** the **authentication data** to **authorized administrators**.

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

a) authorized administrators; and

b) users, which are allowed to modify their own authentication data

5.1.5.12 Management of RBAC TSF Data (FMT_MTD.1(5)) (LSPP/RBAC mode only)

FMT_MTD.1.1 The TSF shall restrict the ability to **modify, create** the **following list of TSF Data** to **a set of RBAC Administrative Roles**:

a) Role Definitions & Role Attributes

b) Role Hierarchies (by assigning one or more roles to other roles)

c) Constraints among Role Relationships

Application Note: [RBACPP] also lists (a) all user passwords, and (e) list auf auditable events; (a) is already covered by FMT_MTD.1(4), and (e) by FMT_MTD.1(2). therefore, these list items have not been repeated here.

5.1.5.13 Secure TSF Data (FMT_MTD.3) (LSPP/RBAC mode only)

FMT_MTD.3.1 The TSF shall ensure that only secure values are accepted for TSF data.

Application Note: This SFR applies to the enforcement of quality metrics for passwords.

5.1.5.14 Revocation of User Attributes (FMT_REV.1(1))

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

FMT_REV.1.2 The TSF shall enforce the rules:

a) The immediate revocation of security-relevant authorizations; and

b) Revocations/modifications made by an authorized administrator to security attributes of a user such as the user identifier, user name, user group(s), user password or user login shell shall be effective the next time the user logs in.

Application Note: Like other UNIX type operating systems also the TOE does not enforce "immediate revocation" for user security attributes. To achieve this, the system administrator has to check, if the user whose security attributes have been changed is currently logged in. If this is the case, the system administrator has to "force" the user to log off as indicated in the Application Note in [LSPP] and [CAPP].

5.1.5.15 Revocation of Object Attributes (FMT_REV.1(2))

FMT_REV.1.1 The TSF shall restrict the ability to revoke security attributes associated with **objects** within the TSC to **users authorized to modify the security attributes by the Discretionary Access Control Policy or *(in LSPP/RBAC mode)* the Mandatory Access Control Policiy and the Role-based Access Control Policy**.

FMT_REV.1.2 The TSF shall enforce the rules:

a) the access rights associated with an object shall be enforced when an access check is made;

b) *(LSPP/RBAC mode)* the rules of the mandatory access control policy are enforced on all future operations;

c) Access rights to file system and IPC objects are checked when the object is opened. Revocations of access rights for file system objects become effective the next time a user affected by the revocation tries to open a file system object or IPC object.

Application Note: Like most other UNIX type operating systems the TOE implements delayed revocation as indicated in the CAPP/LSPP Application Note.

5.1.5.16 Specification of Management Functions (FMT_SMF.1)

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

- **Object security attributes management**
- **User attribute management**
- **Authentication data management**
- **Audit event management**

Application Note: This security functional requirement is not included in the protection profiles and was added because a dependency from FMT_MSA.1 and FMT_MTD.1 to this new component has been defined in [CC].

5.1.5.17 Security Roles (FMT_SMR.2)

FMT_SMR.2.1 The TSF shall maintain the ***following*** roles:

a) authorized administrator; *in LSPP/RBAC mode, these are administrative users assigned to RBAC Administrative Roles*:

- ***(LSPP/RBAC mode) system administrator***
- ***(LSPP/RBAC mode) security administrator***

b) Roles for Object Owners: users authorized by the Discretionary Access Control Policy to modify object security attributes;

c) *(LSPP/RBAC mode)* users authorized by the Mandatory Access Control Policy to modify object security attributes;

d) users authorized to modify their own authentication data; and

e) *no other roles*

FMT_SMR.2.2 The TSF shall be able to associate users with roles.

FMT_SMR.2.3 ***(LSPP/RBAC mode)*** The TSF shall ensure that the ***following*** conditions **for (a) Roles of Object Owners and (b) the set of RBAC Administrative Roles** are satisfied:

a) Object Owners can modify security attributes for only the objects they own

b) The set of RBAC Administrative Roles can modify security attributes for all objects under the control of the TOE.

Application Note: The role model supported by the TOE in CAPP mode is a very simple one: the administrative user is root (extended to all members of the "wheel" group that may su to root). All other users of the system have the user role. As users, they may own certain objects, letting them assume the Owner role for these objects.

In LSPP/RBAC mode, administrative powers are split into different roles as defined by the SELinux policy. By default roles for the system administrator, security administrator, and ordinary users are installed.

5.1.6 Protection of the TOE Security Functions (FPT)

5.1.6.1 Abstract Machine Testing (FPT_AMT.1)

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.

Application Note: The abstract machine testing tool will be platform dependent. Chapter 6 describes the common feature of all those tools.

5.1.6.2 Failure with preservation of Secure State (FPT_FLS.1)(LSPP/RBAC mode only)

FPT_FLS.1.1 The TSF shall preserve a secure state when the following failures occur:

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

5.1.6.3 Manual Recovery (FPT_RCV.1) (LSPP/RBAC mode only)

FPT_RCV.1.1 After **a failure or service discontinuity,** the TSF shall enter a maintenance mode where the ability to return the TOE to a secure state is provided.

5.1.6.4 Function Recovery (FPT_RCV.4) (LSPP/RBAC mode only)

FPT_RCV.4.1 The TSF shall ensure that **the following SFs and failure scenarios** have the property that the SF either completes successfully, or for the indicated failure scenarios, recovers to a consistent and secure state:

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

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

c) no other scenarios

5.1.6.5 Reference Mediation (FPT_RVM.1)

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

5.1.6.6 Domain Separation (FPT_SEP.1)

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

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

Application Note: The TOE enforces this requirement by using the address separation features provided by the Memory Management Units and the protection offered by a multi-state CPU. Although the TOE operates on three different platforms, all those platforms have in common a Memory Management Unit allowing to define address space separation between trusted and untrusted subjects and all platforms support a multi-state CPU where modification to the address space definition and direct access to peripheral devices and the CPU configuration can be restricted to a state reserved for a defined part of the TSF (the kernel). The TOE ensures that those features are used correctly to prohibit any untrusted subject from unallowed interference and tampering with the TSF.

5.1.6.7 Reliable Time Stamps (FPT_STM.1)

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

Application Note: The TOE uses a hardware timer to maintain its own time stamp. This hardware timer is protected from tampering by untrusted subjects. The start value for this timer may be set by the system administrator, but the system administrator may also start a program that uses an external trusted time source to set this initial value.

5.1.6.8 Inter-TSF basic TSF data consistency (FPT_TDC.1) (LSPP mode only)

FPT_TDC.1.1 The TSF shall provide the capability to consistently interpret ***sensitivity labels associated with subjects or objects*** when shared between the TSF and another trusted IT product.

FPT_TDC.1.2 The TSF shall use ***the rules and attributes defined in the MAC SFP*** when interpreting the TSF data from another trusted IT product.

Application note: This requirement is required as a dependency from FDP_ITC.2. Although FDP_ITC.2 is included in LSPP, this dependency has been neither resolved nor has been any rationale provided as to why this dependency does not apply for LSPP. Because the authors of this Security Target do not have access to the evaluation technical report of the LSPP evaluation, the authors of this Security Target do not know if there was a reason for not resolving this dependency. The authors of this Security Target would have expected in any case that the rationale in LSPP provide an explanation why the dependency has not been resolved.

Inter-TSF data consistency shall ensure that sensitivity labels are consistently interpreted when this information is shared between different instantiations of the TOE or when UNIX file system objects with their extended attributes are exported from one system and imported into another system. In order to do this, the definition of the sensitivity labels between the systems involved have to be identical. It is anticipated that the

administrators of the cooperating system share the required definitions. There is no automated mechanism (like a central security database) foreseen to achieve this.

5.1.6.9 TSF Self Test (FPT_TST.1) (LSPP/RBAC mode only)

FPT_TST.1.1 The TSF shall run a suite of self tests **at the request of the authorised user** to demonstrate the correct operation of **the TSF**.

FPT_TST.1.2 The TSF shall provide authorised users with the capability to verify the integrity of **TSF data**.

FPT_TST.1.3 The TSF shall provide authorised users with the capability to verify the integrity of stored TSF executable code.

5.1.7 TOE Access (FTA)

5.1.7.1 Limitation on Scope of Selectable Attributes (FTA_LSA.1) (LSPP/RBAC mode only)

FTA_LSA.1.1 The TSF shall restrict the scope of the session security attributes **(Active Role Set for the User)** based on **the set of Authorized Roles for the User**.

5.1.7.2 TOE Session Establishment (FTA_TSE.1) (LSPP/RBAC mode only)

FTA_TSE.1.1 The TSF shall be able to deny session establishment based on **the default active role set for the user being empty.**

5.1.8 Trusted Path/Channels (FTP)

5.1.8.1 Inter-TSF trusted channel (FTP_ITC.1)

FTP_ITC.1.1 The TSF shall provide a communication channel between itself and a remote trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.

FTP_ITC.1.2 The TSF shall permit ***the TSF or the remote trusted IT product*** to initiate communication via the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for ***when the communication uses the SSH v2 or SSL v3 protocol offered as services by the TOE***.

5.1.9 Strength of Function

The claimed minimum strength of function is *SOF-medium*.

Note: The security function within the TOE that uses a permutational or probabilistic mechanism is the authentication function that uses passwords. No strength of function analysis is performed for the cryptographic algorithms themselves which also excludes any analysis of the existence and characterization of cryptographically weak keys.. This statement is made in compliance with part 1 of the CC and paragraph 414 of part 2 of the CEM.

5.2 TOE Security Assurance Requirements

The target evaluation assurance level for the product is EAL4 [CC] augmented by ALC_FLR.3.

5.3 Security Requirements for the IT Environment

No security functional requirements for the IT environment are applicable, because all security functions are completely implemented without any support from the IT environment.

5.4 Security Requirements for the Non-IT Environment

All the security objectives for the TOE environment address physical protection of the TOE or procedures that need to be obeyed by administrative users.

6 TOE Summary Specification

6.1 Security Enforcing Components Overview

6.1.1 Introduction

This chapter describes the security functions of Red Hat Enterprise Linux that are subject to this evaluation. A large subset of the overall security related functions of Red Hat Enterprise Linux has been included in this evaluation. Those functions provide the basic security for a server within a protected environment. They allow for identification and authentication of users, access control to files and IPC objects, auditing of security critical events and the secure communication with other trusted systems. The TOE protects the security functions from unauthorized tampering and bypassing and allows only administrative users to manage the security functions. Normal users are only allowed to manage access control rights of the file system and IPC objects they own and to modify their own password in accordance with the password rules enforced by the TOE. All other system administration tasks can only be performed by administrative users. In LSPP/CAPP mode, privileges required to perform administrative tasks have been grouped to certain roles to which administrative users can be assigned. In CAPP mode, all privileges are concentrated in the superuser account.Those functions are required as a basis for application level security functions and mechanisms and can be used to build application-specific security policies.

6.1.2 SELinux

The major difference between the CAPP mode of operation and the LSPP/RBAC mode of operation is the activation of the SELinux enhancements with the strict policy which includes MLS restrictions in the latter mode. CAPP mode allows SELinux to be enabled with a policy that does not restrict the capabilities of root, like the targeted policy.

This section discusses the basic architecture and the policies implemented by SELinux using the strict policy for LSPP mode.

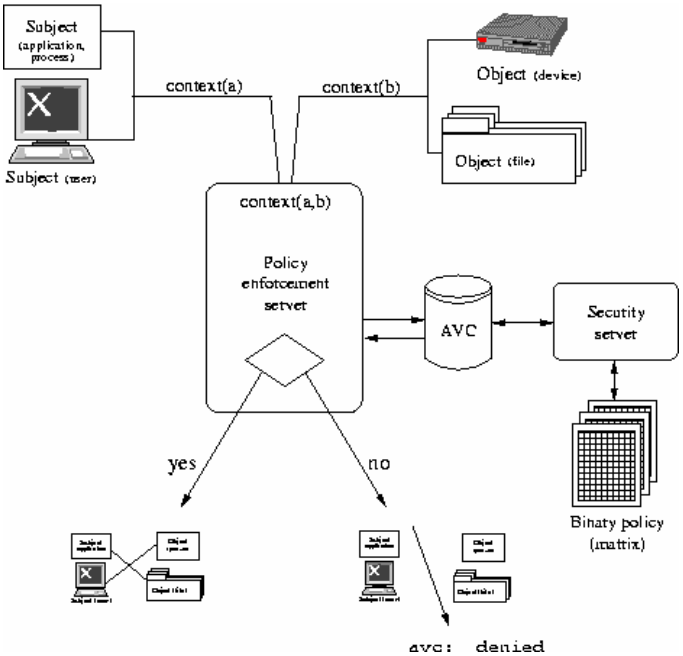
6.1.2.1 SELinux architecture and implementation

Security-enhanced Linux (*SELinux*) is an implementation of a *mandatory access control* mechanism. This mechanism is in the Linux kernel, checking for allowed operations after standard Linux *discretionary access controls* are checked. SELinux is implemented in the Linux kernel using the LSM (*Linux Security Modules*) framework.

To support fine-grained access control, SELinux implements two technologies: *Type Enforcement*™ (TE) and a kind of *role-based access control* (RBAC)

Mandatory and role-based access control using type enforcement is described in Figure 1:

Figure 1: SELinux architecture



A request of a subject (process) to peform an operation (like open, create, write) on an object is routed to the SELinux policy enforcement server, which gathers the security contexts of the subject object and and routes the information to the security server, which and consults the policy database whether the requested operation is allowed or not. The security server provides its decision back to the policy enforcement server, which then enforces this decision by allowing or denying the operation. By separating the security server and its policy database from the policy enforcement server, a variety of policies can be defined without changing the kernel code. For performance reasons, policy decisions are cached in the access vector cache (AVC). All servers are part of the operating system kernel.

Security contexts are a set of four security attributes associated with a process or an object:

user An SELinux user account associated with a subject (denoting the the user on whose behalf the process is executing) or object (the object owner). Note that SELinux user identities are different from the user IDs in the Linux password database.

role Users are authorized to one or more roles, each of which defines a set of privileges or permissions a user can be granted. Every subject can be in only one role at any given time. Transition between roles is accomplished with the *newrole* command, which is somewhat similar to the su command for changing Linux user IDs.

type Types (the term is used interchangeably with the term *domains*) divide subjects and objects into related groups. A type or domain can be thought of as a name for a sandbox in which processes with similar security characteristics execute and in which certain objects are available to them. Currently, over 150 types are defined by a default SELinux installation.

label a sensitivity label consists of a sensitivity range and a set of categories. the TOE provides support for 16 sensitivity levels (s0 to s15) and 256 categories (c0 to c255)

Security contexts have the format of `<user>:<role>:<type>:<label>`

The sequence of access checks is as follows

a) Discretionary access control

b) access control checks based on “traditional” SELinux contexts (user, role, type)

c) access control checks based on sensitivity labels

If one check fails, access is denied without performing the remaining access checks.

6.1.2.2 SELinux policy overview

Policy is the set of rules that guide the SELinux security engine. It defines types for file objects and domains for processes, uses roles to limit the domains that can be entered, and has user identities to specify the roles that can be attained. A domain is what a type is called when it is applied to a process.

A type is a way of grouping together like items based on their fundamental security sameness. This doesn't necessarily have to do with the unique purpose of an application or the content of a document. For example, an object such as a file can have any type of content and be for any purpose, but if it belongs to a user and lives in that user's home directory, it is considered to be of a specific security type, `user_home_t`.

These object types gain their sameness because they are accessible in the same way by the same set of subjects. Similarly, processes tend to be of the same type if they have the same permissions as other subjects. In the targeted policy, programs that run in the unconfined_t domain have an executable with a type such as sbin_t. From an SELinux perspective, that means they are all equivalent in terms of what they can and cannot do on the system.

The policy defines various rules that say how each domain may access each type. Only what is specifically allowed by the rules is permitted. The policy is compiled into binary format for loading into the kernel security server, and as the security server hands out decisions, these are cached in the access vector cache (AVC) for performance.

The SELinux policy includes rules for two different types of decisions:

- access decisions are taken when subjects request access to an object to perform a specific operation, like opening a file for reading or writing. These decisions are based on the current security contexts of the subject and object and the rules defined in the policy and do not change the security context
- transition decisions are required when the security context of a subject or object shall be changed, or when a newly created subject or object needs to be provided with an initial security context.

6.1.2.3 SELinux roles

The following roles are defined in the TOE:

sysadm_r The system administrator role sysadm_r allows the configuration of the SELinux mechanism, including modification of labels (MAC override), configuration and review of audit. The system administrator role sysadm_r usually runs at the lowest sensitivity level (SystemLow) and has MAC override privileges.

staff_r The staff_r role contains all users which have the right to change to the sysadm_r, auditadm_r and secadm_r roles. This allows the prevention of the login of user with immediate roles of secadm_r, auditadm_r and sysadm_r. users must have the respective secadm_r, auditadm_r or sysadm_r role in their set of allowed roles to be able to change to it.

auditadm_r The audit administrator role auditadm_r allows the configuration of the audit subsystem as well as the review of the audit trails.

user_r Normal unprivileged users are assigned to the user_r role. This role does not allow the use of any security relevant mechanism.

6.1.3 Kernel Services

The Red Hat Enterprise Linux kernel includes the base kernel and some kernel modules. The base kernel includes support for system initialization, memory management, file and I/O management, process control, and Inter-Process Communications (IPC) services. Kernel modules are dynamically loadable modules that the kernel will load on demand and that execute with kernel privileges.

Device drivers may be implemented as kernel modules.

The Red Hat Enterprise Linux kernel implements a virtual memory manager (VMM) that allocates a large, contiguous address space to each process running on the system. This address space is spread across physical memory and paging space on a secondary storage device.

The process management component includes the software that is responsible for creating, scheduling, and terminating processes and process threads. Process management allows multiple processes to exist simultaneously on a computer

and to share usage of the computer's processor(s). A process is defined as a program in execution, that is, it consists of the program and the execution state of the program.

Process management also provides services such as inter-process communications (IPC) and event notification. The base kernel implements

- named pipes
- unnamed pipes
- signals
- semaphores
- shared memory
- message queues
- Internet domain sockets
- UNIX domain sockets

Linux supports processes sending signals to each other. However, this is generally limited to processes running under the same user id. The only exception is a process run by the root user: it may send signals to any other process. No DAC checks are involved here. Since the concept of ownership is more restrictive than a role, no RBAC checks are involved either. Hence, signals will only be considered in terms of MAC during this evaluation.

The file and I/O software provides access to files and devices. The Red Hat Enterprise Linux Virtual File System (VFS) provides a consistent view of multiple physical file system implementations. There are the following different types of file systems included in the evaluated configuration: the journalled file system ext3, VFAT filesystem for the /boot/efi partition on Itanium systems (mounted as read/writeable by root only), CDROM/DVD File System ISO-9660 (read-only), and the proc file system, the sysfs file system, the tmpfs file system, the devpts file system, the rootfs file system, the binfmt_misc file system, and the selinuxfs file system.

ext3, VFAT, and ISO-9660 are file systems to be used on a physical medium (disk, CDROM).

The proc and the sysfs file systems do not represent or provide a physical data storage file system but are used as a configuration and monitoring interface to the kernel, provided by the kernel only in a running system. procfs also represents the abstraction of processes (tasks) being files. Processes / tasks are listed as files and directories containing live status information for each process in the system. Process access decisions are enforced by DAC attributes inferred from the underlying process' DAC attributes. The sysfs file system is a new harmonized interface to non-process related kernel information (mainly for device drivers). Access to objects there can be restricted using the DAC mechanism (which are the permission bits only).

The tmpfs file system implements a fast, RAM based file system used for fast access to temporary files. This file system is not persistent across boots of the operating system.

The devpts file system provides pseudo terminal support.

The rootfs file system is used only temporarily during system startup as a placeholder before the real root file system is mounted. It is inaccessible after system startup is complete.

The binfmt_misc file system can be used to register interpreters for executable files based on the file header information, for example to execute Java files directly using the *exec* system call instead of the equivalent invocation of the *java* interpreter with the Java file provided as an argument.

The selinuxfs file system is used for configuration of the SELinux subsystem. It provides an interface to read and set parameters of the SELinux kernel module.

6.1.4 Non-Kernel TSF Services

The non-kernel TSF services are:

- Identification and Authentication services
- Network application layer services

- Configuration and management commands requiring administrative privileges and (in LSPP/RBAC mode) appropriate security contexts.

Those services support the security functions implemented within the kernel and use the kernel interface for this purpose, but they are not running themselves in kernel mode. Those functions are included in the TSF as far as they are required for the security services of the TOE (Identification and Authentication services), while other services that are implemented as tools or commands for the use of the administrative user and where the kernel prohibits the use misuse of those tools or commands since they use kernel functions restricted to administrative users and attempted use by normal users is prohibited by the kernel.

6.1.5 Network Services

The TOE is capable of providing the following types of services:

- Local services to the user currently logged in to the local computer console.
- Local services to previous users via deferred jobs.
- Local services to users who have accessed the local host via the network using protocols such as ftp or ssh.
- Network services to clients on either the local host or on remote hosts.

Network services are provided to clients via a client-server architecture. This client-server architecture refers to the division of the software that provides a service into a client portion, which makes requests, and a server portion, which carries out client requests (usually on a different computer). A service protocol acts as the interface between the client and server.

The primary low-level protocols are Internet Protocol (IP), Transmission Control Protocol (TCP), and User Datagram Protocol (UDP). IP is not user visible, but non-TSF processes may communicate with other hosts in a networked system using a reliable byte stream or unreliable datagrams, TCP and UDP respectively.

The higher-level network services are built on TCP or UDP. The TCP based application protocols supporting user authentication and running on privileged ports are:

- secure shell (SSH v2)
- file transfer services (FTP)

In addition the TOE supports secure socket layer (SSL v3) protocol, which can be used to securely tunnel higher layer protocols. This service is provided by a trusted process which can be used by applications to tunnel TCP based protocols using a single port. The tunnel actually provides the certificate based authentication of the server side of the tunnel and the confidentiality and integrity protection of the communication.

6.1.6 Security Policy Overview

The TOE is a single Red Hat Enterprise Linux system running on one machine. Several of those systems may be interconnected via a local area network and exchange information using the network services. But one should keep in mind that the following statements hold:

- There is a Linux (Red Hat Enterprise Linux) kernel running on each host computer in the networked system.
- Identification and authentication (I&A) is performed locally by each host computer. Each user is required to log in with a valid password and user identifier combination at the local server and also at any remote computer where the user can enter commands to a shell program (using ssh). User ID and password for one human user may be different on different hosts. User ID and password on one host system are not known to other host systems on the network and therefore a user ID is relevant only for the host where it is defined.
- Discretionary access control (DAC) is performed locally by each of the host computers and is based on user identity and group membership on this host. Each process has an identity (the user on whose behalf it is operating) and belongs to one or more groups. All named objects have an owning user, an owning group and a DAC attribute, which is a set of permission bits. In addition, file system objects optionally have extended permissions also known as an Access Control List (ACL). The ACL mechanism is a significant enhancement

beyond traditional UNIX systems, and permits control of access based on lists of users and/or groups to whom specific permissions may be individually granted or denied.

- (LSPP/RBAC only) Mandatory access control (MAC): MAC is performed locally by each of the host computers and is based on the labels of a process (representing a subject) and the labels of the object the user tries to access. Each process and each object is maintained with its label.
- (LSPP/RBAC only) Role-based access control (RBAC): The system implements the role based access control effectively limiting the possibilities of a subject to access different objects (such as executing applications). This mechanism removes the all-powerful root user by limiting the root user's capabilities to the capabilities allowed by the associated role.
- Object reuse is performed locally, without respect to other hosts.
- Interrupt handling is performed locally, without respect to other hosts.
- (CAPP mode): Privilege is based on the root identity. All privileged processes (SUID root programs and programs run under the root identity) start as processes with all privileges enabled. Unprivileged processes, which include SGID trusted processes, start and end with no privileges enabled.

6.1.7 TSF Structure

The TSF is the portion of the system that is responsible for enforcing the system's security policy. The TSF of Red Hat Enterprise Linux consists of two major components: kernel software and trusted processes. All these components must operate correctly for the system to be trusted. Those functions are supported by the mechanisms of the underlying hardware which are used to protect the TSF from tampering by untrusted processes.

The Red Hat Enterprise Linux hardware platforms support two execution states where kernel mode or supervisor state, software runs with hardware privilege and user mode or problem state software runs without hardware privilege. Red Hat Enterprise Linux also provides two types of memory protection: segmentation and page protection. The memory protection features isolate critical parts of the kernel from user processes and ensure that segments in use by one process are not available to other processes. The two-state architecture and the memory protections form the basis of the argument for process isolation and protection of the TSF.

The trusted processes include programs such as Linux administrative programs, scripts, shells, and standard Linux utilities that run with administrative privilege, as a consequence of being invoked by a user with administrative privileges. Non-kernel TSF software also includes daemons that provide system services, such as networking, as well as SUID and SGID programs that can be executed by untrusted users.

6.1.8 TSF Interfaces

Each subsection here summarizes a class of interfaces in the Red Hat Enterprise Linux operating system, and characterizes them in terms of the TSF boundary. The TSF boundary includes some interfaces, such as commands implemented by privileged processes, which are similar in style to other interfaces that are not part of the TSF boundary and thus not trusted. Some interfaces are part of the TSF boundary only when used in a privileged environment, such as an administrative user's process, but not when used in a non-privileged environment, such as a normal user process. All interface classes are described in further detail in the next chapter, and the mechanisms in subsequent chapters. As this is only an introduction, no explicit forward references are provided.

6.1.8.1 User Interfaces

The typical interface presented to a user is the command interpreter, or shell. The user types commands to the interpreter, and in turn, the interpreter invokes programs. The programs execute hardware instructions and invoke the kernel to perform services, such as file access or I/O to the user's terminal. A program may also invoke other programs, or request services using an IPC mechanism. Before using the command interpreter, a user must log in.

The command interpreter or shell as well as other programs operating on behalf of a user have the following interfaces:

- CPU instructions, which a process uses to perform computations within the processor's registers and a process's memory areas. CPU instructions are interpreted by the hardware, which is part of the TOE. CPU instructions are therefore an interface to the TOE.

- System calls (e.g. open, fork), through which a process requests services from the kernel, which are invoked using a special CPU instruction. System calls are the primary way for a program operating on behalf of a user to request services of the TOE including the security services. System calls related to security functions are therefore part of the TSF interface.
- Directly-invoked trusted processes (e.g. passwd) which perform higher-level services, and are invoked with an exec system call that names an appropriate program which is part of the TSF, and replaces the current process's content with it; a limited number of those processes exist that perform security functions and are therefore part of the TSF interface.
- Daemons, which accept requests stored in files or communicated via other IPC mechanisms, generally created through use of directly invoked processes (some trusted, some untrusted). A few daemons perform security functions and therefore part of the TSF interface.
- Network Services, (ssh, ftp, ssl). The network services interface operates at many different levels of abstraction. At the highest level, it provides a means for users on one host to request a virtual terminal connection on another host within the system. At a lower level, it allows a host on a networked system to request a specific service from another host within the system on behalf of a user. Examples of requested services include remotely login into the TOE and obtaining a shell or transferring whole files. At the lowest level, it allows a subject on one host in the system to request a connection (i.e. TCP), or deliver data (i.e. UDP) to a listening subject. Network services usually consist of a client on the requestor's side and a server (usually a daemon) running on the server's side. Authentication (if required by the service) and access control use dedicated interfaces to the functions on the server side which are therefore part of the TSF interface. Note that for the TOE only ssh, ssl and ftp are seen as TSF, because they use privileged ports. ssh and ftp require user identification and authentication and ssh and ssl provide confidentiality and integrity protection

  Note: Users may start programs using unprivileged ports, but those programs operate with the effective and filesystem userid of the calling user and are therefore restricted by the security policy of the TOE. Those user programs using unprivileged ports are not part of the TSF.

6.1.8.2 Operation and Administrator Interface

The primary administrative interfaces to Red Hat Enterprise Linux are the same as the interfaces for ordinary users; the administrative user logs into the system with a standard, untrusted, identity and password, and after assuming the root identity uses standard Linux commands to perform administrative tasks. Direct root login is only allowed from the system console (to avoid a denial of service attack).

The part of the administrative database (which is the set of all security relevant configuration files) that is used to configure and manage TSF is seen as part of the TSF interface. The administrative database is protected by the access control mechanisms of the TOE. It is therefore very important to set the access rights to the files of the administrative database such that non-administrative users are prohibited from modifying those files and have read access on a need to know basis only. Note that each server in the system has its own administrative database and if synchronization between those TSF database is required by the organization's security policy, it has to be done manually in the system environment. The TOE does not provide any function to synchronize TSF databases on different systems.

6.1.9 Secure and Non-Secure States

The secure state for the Red Hat Enterprise Linux is defined as a host's entry into multi-user mode with the administrative databases configured with the required access rights. At this point, the host accepts user logins and services network requests across the networked system. If these facilities are not available, the host is considered to be in a non-secure state. Although it may be operational in a limited sense and available for an administrative user to perform system repair, maintenance, and diagnostic activity, the TSF are not in full operation and are not necessarily protecting all system resources according to the security policy.

6.2 Description of the Security Enforcing Functions

6.2.1 Introduction

This chapter describes how the Security Enforcing components of the TOE provide the Security Requirements identified in chapter 5.

A high level description is provided for each group of security enforcing functions (SEF) providing a common feature or service, and stating how the functionality specified by the security enforcing function group is provided by the security enforcing components identified in this Chapter.

The security enforcing function groups identified in this chapter follow the description given in chapter 2:

- Identification and Authentication
- Audit
- Discretionary Access Control
- Mandatory Access Control
- Role-based Access Control
- Object Reuse
- Security Management
- Secure Communication
- TOE Protection

The TOE security functions (TSF) are described with sufficient detail to provide a general understanding of those functions and how they work. A more detailed description of those functions and a mapping of the TSF to TOE subsystems is provided in the high level design documentation.

References to components given in *italics* can be traced to manual pages or TOE sources for further information. Note also that some commands initiate trusted processes or are a local front end to a trusted process (e.g. *ftp* and the *ftpd* daemon, *ssh* and the *sshd* daemon). In these instances, a generic reference to the command is made.

6.2.2 Identification and Authentication (IA)

User identification and authentication in the Red Hat Enterprise Linux includes all forms of interactive login (e.g., using the *ssh* or *ftp* protocols) as well as identity changes through the *su* command. These all rely on explicit authentication information provided interactively by a user.

Identification and authentication of users is performed from a terminal where no user is logged on or when a user that is logged on starts a service that requires additional authentication. All those services use a common mechanism for authentication described in this chapter. They all use the administrative database. The administrative database is managed by administrative users, but normal users are allowed to modify their own password using the *passwd* command. This chapter also describes the authentication process for those network services that require authentication.

Linux uses a suite of libraries called the „Pluggable Authentication Modules" (PAM) that allow an administrative user to choose how PAM-aware applications authenticate users. This section provides also a brief description how PAM is used and configured in the evaluated configuration:

- The evaluated configuration supports password based login only (pam_unix.so module). To restrict the use of the *su* command to members of the "wheel" group the pam_wheel.so module is used.
- The module pam_rootok.so allows a user with an effective userid of 0 to use several administrative commands without re-authentication.
- The module pam_tally.so counts the number of consecutive unsuccessful authentication attempts for a user and blocks further login attempts for this user until an administrative user unblocks the user.
- The module pam_securetty.so is used to restrict the login of root to a terminal listed in */etc/securetty*.

- The module pam_nologin.so is used to allow restricting login to root only (for example when critical system management activities need to be performed). If the file */etc/nologin* exists, the TOE rejects login attempts from any user except root and displays the message found in the file */etc/nologin* to users that try to log into the TOE.
- The module pam_passwdqc.so provides additional checks for the strength of passwords, allowing for a stricter password policy.
- The module pam_loginuid.so sets the user identification used by audit, and provides fail secure mode by preventing user login if the audit subsystem is inoperative.
- LSPP/RBAC mode: The module pam_selinux.so sets the security context for the initial subject, including the sensitivity label.

6.2.2.1 User Identification and Authentication Data Management (IA.1)

Each server maintains its own set of users with their passwords and attributes. Although the same human user may have accounts on different servers interconnected by a network and running an instantiation of the TOE, those accounts and their parameter are not synchronized on different servers. As a result the same user may have different usernames, different user IDs, different passwords and different attributes on different machines within the networked environment. Existing mechanisms for synchronizing such information within the whole networked system are not subject to this evaluation.

Each machine within the network maintains its own administrative database by making all administrative changes on the local machine. System administration has to ensure that all machines within the network are configured in accordance with the requirements defined in this Security Target.

Users are allowed to change their own passwords by using the *passwd* command. The password command bears the privileges (setuid to 0 in CAPP mode, in addition the appropriate security context in LSPP/RBAC mode) to access and modify the user authentication database (*/etc/shadow*). If invoked by unprivileged users, it is trusted to change only the password of the user who invoked it and only after the user authenticated properly; administrative users are allowed to change the password of arbitrary users (IA.1.1).

Users are also forced to change their passwords at login time, if the password has expired (IA1.2).

The file */etc/passwd* contains the user's name, the id of the user, an indicator, if the password of the user is valid, the principal group id of the user and a few other, not security relevant attributes (IA1.3). The hashed password of the user itself is not stored in this file but in the file */etc/shadow* which can be protected against read access for ordinary users. This prohibits dictionary attacks on passwords in the passwd file.

The file */etc/shadow* contains the MD5 hashed password, the userid, the time the password was last changed and some other information that are not subject to the security functions as defined in this Security Target (IA1.4).

For a complete list of user attributes see the description of the function SM.3.

An administrative user can define the following restrictions on the login process (defined in /etc/login.defs to be used by management tools; in the PAM configuration and the trusted database */etc/shadow* to be used by the authentication process itself):

- Maximum number of days a password may be used.
- Minimum number of days allowed between password changes.
- Minimum acceptable password length (defined in the parameter to pam_passwdqc.so).
- Number of days a warning is given before a password expires.
- Number of consecutive unsuccessful login retries.
- Number of old but recent passwords to be disallowed when changing the password for a user (password history)

This allows an administrative user to define restrictions on authentication data (IA1.5). Those restrictions are stored in the file */etc/login.defs*, */etc/shadow* and in the PAM configuration. The administrative user can use those parameters to define a password policy such that the passwords satisfy the requirements defined in FIA_SOS.1.

The time of the last successful logins is recorded in /var/log/lastlog (IA1.6).

In the evaluated configuration the above mentioned parameter need to be set in accordance with the following restrictions (IA1.7):

- Maximum lifetime of a password: less than or equal to 60 days
- Minimum lifetime of a password: 1 day
- Minimum length of a password: 8 character
- Number of days a warning is given before password expires: 7 days
- Number of consecutive unsuccessful login retries: 5
- Password history length: 7

This function contributes to satisfy the security requirements FIA_ATD.1, FIA_SOS.1, FMT_MSA.2, FMT_MTD.1(3), FMT_MTD.3 and FMT_SMF.1.

6.2.2.2 Common Authentication Mechanism (IA.2)

Red Hat Enterprise Linux includes a common authentication mechanism which is a subroutine used for all activities that create a user session, including all the interactive login activities, batch jobs, and authentication for the *su* command (IA2.1).

The common mechanism includes the following checks and operations:

- Check password authentication
- Check password expiration
- Check whether access should be denied due to too many consecutive authentication failures

The common I&A mechanism identifies the user based on the supplied user name and performs authentication against the user's password. (IA.2.2)

This function contributes to satisfy the security requirements FIA_UAU.2 and FIA_UID.2.

6.2.2.3 Interactive Login and Related Mechanisms (IA.3)

The *ssh* and *ftp* as well as the *su* command used to change the real, filesystem, and effective user ID of a user all use the same authentication mechanism in the evaluated configuration (IA3.1). It is of course up to the remote system to protect the user's entry of a password correctly (e. g. provide only obscured feedback). As long as the remote system is also an evaluated version of the TOE, this is ensured by the security function of the TOE.

This function contributes to satisfy the security requirements FIA_UAU.2, FIA_UID.2 and FIA_UAU.7.

6.2.2.4 User Identity Changing (IA.4)

Users can change their identity (i.e., switch to another identity) using the *su* command (IA4.1). When switching identities, the real, filesystem, and effective user ID and real, filesystem, and effective group ID are changed to the one of the user specified in the command (after successful authentication as this user) (IA4.2). The primary use of the *su* command within the Red Hat Enterprise Linux is to allow appropriately authorized individuals the ability to assume the root identity to perform administrative actions. In this system the capability to login as the root identity has been restricted to defined terminals only (IA4.3). In addition the use of the *su* command to switch to root has been restricted to users belonging to the "wheel" group (IA4.4). Users that don't have access to a terminal where root login is allowed and are not member of the "wheel" group will not be able to switch their real, filesytem, and effective user ID to root even if they would know the authentication information for root. Note that when a user executes a program that has the SUID bit set only the effective user ID and filesystem ID are changed to that of the owner of the file containing the program while the real user ID remains that of the caller (IA4.5). The login ID is neither changed by the *su* command nor by executing a program that has the SUID or SGID bit set (IA4.6).

The *su* command invokes the common authentication mechanism to validate the supplied authentication.

This function contributes to satisfy the security requirement FIA_USB.1.

6.2.2.5 Login Processing (IA.5)

At the login process the login, real, filesystem, and effective user ID are set to the ID of the user that has logged in (IA5.1).

In LSPP/RBAC mode, the login processing sets the initial security context, including SELinux user identity, role, type and sensitivity label (see MA.1 and RBAC.1) (IA.5.2).

With the *su* command the real, filesystem, and the effective user ID and the real, filesystem, and the effective group ID are changed but the login ID remains unchanged. (IA5.3)

This function contributes to satisfy the security requirement FIA_USB.1.

TOE access (IA.6)

The TOE restricts the active role set for the user to the set of authorized roles assigned to that user at user-subject binding time (during interactive login or when a batch job is initiated) (IA.6.1). The initialization of a session for a user is denied if the set of authorized rules for the user is empty (IA.6.2).

This function contributes to satisfy the security functional requirements FTA_LSA.1 and FTA_TSE.1.

6.2.3 Audit (AU)

The Lightweight Audit Framework (LAF) is designed to be an audit system for Linux that complies with the requirements set forth in CAPP, LSPP and RBAC. The subsystem allows configuring the events to be actually audited from the set of all events that are possible to be audited. Those events are configured in a specific configuration file and then the kernel is notified to build its own internal structure for the events to be audited.

6.2.3.1 Audit Configuration (AU.1)

An administrative user (in LSPP/RBAC mode in the secadm_r role) can define the events to be audited from the overall events that the Lightweight Audit Framework is able to audit using rules defined in the */etc/audit.rules* audit configuration file using simple filter expressions (AU1.1). This allows for a flexible definition of the events to be audited and the conditions under which events are audited. The system administrator is also able to define a set of user IDs for which auditing is active (AU1.2) or alternatively a set of user IDs that are not audited (AU1.3). Changes to the audit configuration take effect when the audit daemon is notified about a change in the audit configuration (AU1.4).

This notification can only be performed by an administrative user (using the */etc/rc.d/init.d/auditd* script with the 'reload' parameter) (AU1.5).

The system adminstrator can select files to be audited by adding them to a watch list that is loaded into the kernel using the *auditctl* tool each time the audit system is started or reinitialized. The list allows the administrator to select an arbitrary audit tag value for each file which will be preserved as a searchable attribute in the audit log (AU1.6). The kernel interface for configuring these audit properties is usable only by root users (AU1.7).

This function contributes to satisfy the security requirements FAU_SEL.1 and FMT_MTD.1(1)

6.2.3.2 Audit Processing (AU.2)

Auditing is performed on a per-process basis. A process can enable or disable auditing for itself by attaching itself or detaching itself to the audit subsystem provided it is running with administrative privileges (AU2.1). The attribute of being attached to the audit subsystem is inherited by all processes that are forked off from a process, which ensures that events generated by child processes are also audited (AU2.2).

The kernel audits system calls in accordance with the rules defined in the *audit.rules* audit configuration file. In addition, trusted processes can generate audit records and send them to the kernel (AU2.3). The login ID is associated with audit events ensuring that events can be easily associated with the ID a user used to log into the TOE (AU2.4).

The events to be audited are forwarded by the kernel to an audit daemon, which writes the audit records to the audit trail. If the file system space does not have sufficient space to store new audit records or a configurable space threshold is exceeded, the TOE switches into single user mode or is halted depending on the configuration of the audit daemon

(AU2.5). This ensures that audit records do not get lost due to resource shortage and the administrator can backup and clear the audit trail to free disk space for new audit logs.

The audit daemon appends audit records to a file whose name is specified in the audit configuration file (AU2.6).

The audit configuration file can be used to execute administrator-specified notification actions when the free disk space available reaches an administrator-specified threshold (AU2.7). This is used to inform the system administrator that he needs to back-up the current audit trail and make space available for additional audit records. In the case the system administrator does not perform this in time and the available disk space is exhausted, the audit daemon can be configured to switch to single user mode or to halt the whole system. In that case the system administrator will need to back-up and clear the audit trail in single user mode and then re-boot the TOE in secure multiuser mode.

Access to audit data by normal users is prohibited by the DAC, MAC and RBAC functionality of the TOE, which is used to restrict the access to the audit trail and audit configuration files to the system administrator only.

This function contributes to satisfy the security requirements FAU_SAR.2, FAU_STG.1, FAU_STG.3, FAU_STG.4, FMT_MTD.1(1) and FMT_SMR.2.

6.2.3.3 Audit Record Format (AU.3)

An audit record consists of one or more lines of text containing fields in a “keyword=value” tagged format. The following information is contained in all audit record lines:

- Type: indicates the source of the event, such as SYSCALL, FS_WATCH, USER, or LOGIN
- Timestamp: Date and time the audit record was generated
- Audit ID: unique numerical event identifier
- login ID (“auid”), the user ID of the user authenticated by the system (regardless if the user has changed his real and / or effective user ID afterwards)
- Effective user ID: the effective user ID of the process at the time the audit event was generated
- Success or failure (where appropriate)

(AU3.1)

This information is followed by event specific data. In some cases, such as syscall event records involving file system objects, multiple text lines will be generated for a single event, these all have the same timestam and audit ID to permit easy correlation.

Note: Although the TOE distinguishes between the effective and the filesystem user ID, those two are identical in all states of the TOE.

The event specific data will always contain data indicating if the request that caused the event has been successful or not (AU3.2).

The audit subsystem maintains a “Login ID” which is set when the user performs his initial login at a terminal or via a network connection (AU3.3). This Login ID is maintained for actions of this user until he terminates the session. This Login ID remains unchanged when the user performs a switch of the real and / or effective and filesystem user ID by the *su* command or by invoking a program that has the SUID bit set (AU3.4). This allows tracing of all actions to the real user.

This function contributes to satisfy the security requirements FAU_GEN.1 and FAU_GEN.2.

6.2.3.4 Audit Post-Processing (AU.4)

The TOE provides tools for managing ASCII files that can be used for post-processing of audit data. These tools include:

ausearch reads the ASCII audit data (AU4.1)

ausearch allows selective extraction of records from the audit trail using defined selection criteria (AU4.2). The human-readable records can be postprocessed further with other tools, e.g. to sort the output according to various criteria (AU.4.3)

This function contributes to satisfy the security requirements FAU_SAR.1 and FAU_SAR.3.

6.2.4 Discretionary Access Control (DA)

This section outlines the general DAC policy in Red Hat Enterprise Linux as implemented for resources where access is controlled by permission bits and POSIX ACLs; principally these are the objects in the file system. In all cases the policy is based on user identity (and in some cases on group membership associated with the user identity). To allow for enforcement of the DAC policy, all users must be identified and their identities authenticated.

Details of the specific DAC policy applied to each type of resource are covered in the section “Discretionary Access Control: File System Objects” and the section “Discretionary Access Control: IPC Objects”.

Note: Signals are not subject to discretionary access control as described in this section of the Security Target. The rules when a process is allowed to send a signal to another process are not seen as security relevant and therefore not listed in this Security Target.

Note: In LSPP/RBAC mode, the TOE supports different access control mechanisms. DAC is always performed first, and all other access checks are only performed if DAC checks succeed. Therefore, MAC and RBAC can only impose additional restrictions over the DAC access rights and never grant more rights than have been specified by the DAC policy. This implies that in LSPP/RBAC mode, the checks against UID 0 (or the set of kernel privileges associated with it) as described in this section for DAC checks are still performed and must succeed for DAC to allow an operation. They may, however, be not sufficient to perform that operation, as the subsequently invoked RBAC and MAC checks may deny the operation based on their policy.

6.2.4.1 General DAC Policy (DA.1)

The general policy enforced is that subjects (i.e., processes) are allowed only the accesses specified by the class-specific policies. Further, the ability to propagate access permissions is limited to those subjects who have that permission, as determined by the class-specific policies.

Finally, in CAPP mode a subject with a filesystem user ID of 0 is exempt from all restrictions and can perform any action desired (DA1.1). In LSPP/RBAC mode, the restrictions of the MAC and RBAC policies additionally apply.

DAC provides the mechanism that allows users to specify and control access to objects that they own (DA1.2). DAC attributes are assigned to objects at creation time and remain in effect until the object is destroyed or the object attributes are changed (DA1.3). DAC attributes exist for, and are particular to, each type of object on Red Hat Enterprise Linux. DAC is implemented with permission bits and, when specified, ACLs.

A subject whose filesystem user ID matches the file owner ID can change the file attributes, the base permissions, and the extended permissions (except for read-only file systems, of course) (DA1.4). Changes to the file group are restricted to the owner and administrative users (DA1.5).

The new file group identifier must either be the current filesystem group identifier or one of the group identifiers in the concurrent group set (DA1.6). In addition, a subject whose filesystem user ID is 0 can make any desired changes to the file attributes, the base permissions, the extended permissions, and owning user of the file (see DA1.1).

Permission bits are the standard UNIX DAC mechanism and are used on all Red Hat Enterprise Linux file system named objects (DA1.7). Individual bits are used to indicate permission for read, write, and execute access for the object’s owner, the object’s group, and all other users (i.e. world). The extended permission mechanism is supported only for file system objects within an ext3 file system and provides a finer level of granularity than do permission bits (DA1.8). The VFAT formatted /boot/efi partition on Itanium systems have partition-wide enforced permission bits which are configured during mount time (this partition must be configured to be accessible by root only).

Write access is in general not granted for files on a file system mounted as read-only (DA1.9). Write access is also denied for files that have the immutable attribute (DA1.10).

This function contributes to satisfy the security requirements FDP_ACC.1(1) and FDP_ACF.1(1).

6.2.4.2 Permission Bits (DA.2)

Red Hat Enterprise Linux supports standard UNIX permission bits to provide one form of DAC for file system objects in the vfat, iso9660, procfs, sysfs, tmpfs, devpts, binfmt_misc, and selinuxfs file systems. (Permissions for the rootfs file system are irrelevant as this file system is not accessible after system startup.)

There are three sets of three bits that define access for three categories of users: the owning user, users in the owning group, and other users. The three bits in each set indicate the access permissions granted to each user category: one bit for read (r), one for write (w) and one for execute (x). Note that write access to file systems mounted as read only (e. g. CD-ROM) is always rejected. Note also that access to specific objects in the /proc file system may be restricted to root regardless of the setting of the permission bits.

Each subject's access to an object is defined by some combination of these bits:

- rwx symbolizing read/write/execute
- r-x symbolizing read/execute
- r-- symbolizing read
- --- symbolizing null
  (DA2.1)

When a process attempts to reference an object protected only by permission bits, the access is determined as follows:

- Users with a filesystem user ID of 0 are able to read and write all files, ignoring the permission bits. Users with a filesystem user ID of zero are also able to execute any file if it is executable for someone.
- If the filesystem user ID = object's owning user ID and the owning user permission bits allow the type of access requested access is granted or denied with no further checks.
- If the filesystem group ID, or any supplementary groups of the process = object's owning group ID, and the owning group permission bits allow the type of access requested access is granted or denied with no further checks.
- If the process is neither the owner nor a member of an appropriate group and the permission bits for world allow the type of access requested, then the subject is permitted access.
- If none of the conditions above are satisfied, and the process is not the root identity, then the access attempt is denied.
  (DA2.2)

It is to be noted that on the VFAT file system, permission bits are set as mount option and apply to the whole partition.

This function contributes to satisfy the security requirements FAU_SAR.2, FDP_ACC.1(1), FDP_ACF.1(1) and FIA_USB.1.

6.2.4.2.1 DAC: SYSFS File System

The 2.6 version of the Linux kernel has introduced a new file system to assist in the handling of device driver charcteristics. The sysfs file system uses the permission bits for access control and protection and therefore access control follows the semantics of the permission bits (DA2.3). A specific type of access requires that a method to handle this access has been registered with the object when it was created in the sysfs file system (DA2.4).

This function contributes to satisfy the security requirements FDP_ACC.1(1), FDP_ACF.1(1), FMT_MSA.1(1), FMT_SMF.1, FMT_MSA.3(1) and FPT_SEP.1.

6.2.4.3 Access Control Lists supported by Red Hat Enterprise Linux (DA.3)

Red Hat Enterprise Linux provides support for POSIX type ACLs for the ext3 file system allowing to define a fine grained access control on a user basis. The semantics of those ACLs is summarized in this section.

An ACL entry contains the following information:

1. A tag type that specifies the type of the ACL entry
2. A qualifier that specifies an instance of an ACL entry type
3. A permission set that specifies the discretionary access rights for processes identified by the tag type and qualifier
(DA3.1)

6.2.4.3.1 ACL Tag Types

The following tag types exist:

1. ACL_GROUP
an ACL entry of this type defines access rights for processes whose filesystem group ID or any supplementary group IDs match the one in the ACL entry qualifier
2. ACL_GROUP_OBJ
an ACL entry of this type defines access rights for processes whose filesystem group ID or any supplementary group IDs match the group ID of the group of the file
3. ACL_MASK
an ACL entry of this type defines the maximum discretionary access rights a process in the file group class
4. ACL_OTHER
an ACL entry of this type defines access rights for processes whose attributes do not match any other entry in the ACL
5. ACL_USER
an ACL entry of this type defines access rights for processes whose filesystem user ID matches the ACL entry qualifier
6. ACL_USER_OBJ
an ACL entry of this type defines access rights for processes whose filesystem user ID matches the user ID of the owner of the file
(DA3.2)

6.2.4.3.2 ACL Qualifier

The qualifier is required for ACL entries of type ACL_GROUP and ACL_USER and contain either the user ID or the group ID for which the access rights defined in the entry shall apply (DA3.3).

6.2.4.3.3 ACL Permissions

The permissions that can be defined in an ACL entry are: read, write and execute/search (DA3.4).

6.2.4.3.4 Relation with File Permission Bits

An ACL contains exactly one entry for each of the ACL_USER_OBJ, ACL_GROUP_OBJ, and ACL_OTHER tag type (called the „required ACL entries") (DA3.5). An ACL may have between zero and a defined maximum number of entries of the type ACL_GROUP and ACL_USER (DA3.6).

An ACL that has only the three required ACL entries is called a „minimum ACL". ACLs with one or more ACL entries of type ACL_GROUP or ACL_USER are called an „extended ACL".

The standard UNIX file permission bits as described in the previous section are represented by the entries in the minimum ACL. The owner permission bits are represented by the entry of type ACL_USER_OBJ, the entry of type ACL_GROUP_OBJ represent the permission bits of the file's group and the entry of type ACL_OTHER represents the permission bits of processes running with a filesystem user ID and filesystem group ID or supplementary group ID different from those defined in ACL_USER_OBJ and ACL_GROUP_OBJ entries (DA3.7).

6.2.4.3.5 ACL_MASK

If an ACL contains an ACL_GROUP or ACL_USER type entry, then exactly one entry of type ACL_MASK is required in the ACL. Otherwise the entry of type ACL_MASK is optional (DA3.8).

6.2.4.3.6 Default ACLs

A default ACL is an additional ACL which may be associated with a directory. This default ACL has no effect on the access to this directory. Instead the default ACL is used to initialize the ACL for any file that is created in this directory. If the new file created is a directory it inherits the default ACL from its parent directory (DA3.9).

When an object is created within a directory and the ACL is not defined with the function creating the object, the new object inherits the default ACL of its parent directory as its initial ACL.

6.2.4.3.7 Access Check Evaluation Algorithm

When a process attempts to reference an object protected by an ACL, it does so through a system call (e.g., open, exec). If the object has been assigned an ACL access is determined as according to the algorithm below:

ACCESS CHECK ALGORITHM
A process may request read, write, or execute/search access to a file system object protected by an ACL. The access check algorithm determines whether access to the object will be granted.

1. Write access to a file on a read-only file system will always be denied for file system objects other than device special files.

2. Write access to a file with the immutable attribute will always be denied.

3. **If** the filesystem user ID of the process matches the user ID of the file object owner, **then**

   if the ACL_USER_OBJ entry contains the requested permissions, access is granted,

   else access is denied.

4. **else if** the filesystem user ID of the process matches the qualifier of any entry of type ACL_USER, **then**

   if the matching ACL_USER entry and the ACL_MASK entry contain the requested permissions, access is granted,

   else access is denied.

5. **else if** the filesystem group ID or any of the supplementary group IDs of the process match the qualifier of the entry of type ACL_GROUP_OBJ, or the qualifier of any entry of type ACL_GROUP, **then**

   if the ACL_MASK entry and any of the matching ACL_GROUP_OBJ or ACL_GROUP entries contain all the requested permissions, access is granted,

   else access is denied.

6. **else if** the ACL_OTHER entry contains the requested permissions, access is granted.

7. **else** access is denied**.**

(DA3.10)

This function contributes to satisfy the security requirement FDP_ACC.1(1), FDP_ACF.1(1) and FIA_USB.1

6.2.4.3.8 DAC Revocation on File System Objects

File system objects access checks are performed when the object is initially opened, and are not checked on each subsequent access. Changes to access controls (i.e., revocation) are effective with the next attempt to open the object (DA3.11).

In cases where an administrative user determines that immediate revocation of access to a file system object is required, the administrative user can reboot the computer, resulting in a close on the object and forcing an open of the object on system reboot.

6.2.4.3.9 DAC: Directory

The execute permission bit for directories governs the ability to name the directory as part of a pathname. A process must have search (execute) access in order to traverse the directory during pathname resolution (DA3.12).

Directories may not be written directly, but only by creating, renaming, and removing (unlinking) objects within them. These operations are considered writes for the purpose of the DAC policy (DA3.13).

6.2.4.3.10 DAC: UNIX Domain Socket Special File

UNIX domain socket files are treated as files in the Red Hat Enterprise Linux file system from the perspective of access control, with the exception that using the bind or connect system calls requires that the calling process must have write access to the socket file (DA3.14).

UNIX domain sockets exist in the file system name space, the socket files can have both base mode bits and extended ACL entries (DA3.15).

UNIX domain sockets consist of a socket special file (managed by the File System) and a corresponding socket structure (managed by IPC). The TOE controls access to the socket based upon the caller's rights to the socket special file (DA3.16).

6.2.4.3.11 DAC: Named Pipes

Named pipes are treated identically to any other file in the Red Hat Enterprise Linux file system from the perspective of access control. Therefore permission bits and extended permissions can be used (DA3.17). For this reason named pipes are listed as file system objects (although they are used for interprocess communication). Note that named pipes follow the rules for IPC objects, if no ACLs are used (which probably is the normal case they are used).

6.2.4.3.12 DAC: Device Special File

The access control scheme described for file system objects is used for protection of character and block device special files (DA3.18). Most device special files are configured to allow read and write access by the root user, and read access by privileged groups. With the exception of terminal and pseudo-terminal devices and a few special cases (e.g., /dev/null and /dev/tty), devices are configured to be not accessible to normal users (DA3.19). The access mode of device files for ttys is changed during login time to read/write access of the user logging into the system; on logout the access rights are reset to allow only access by root (DA3.20).

This function contributes to satisfy the security requirement FDP_ACC.1(1), FDP_ACF.1(1), FIA_USB.1 FMT_MSA.1(1), FMT_SMF.1, FMT_MSA.3(1), FPT_SEP.1 and FMT_REV.1(2).

6.2.4.4 Discretionary Access Control: IPC Objects (DA.4)

6.2.4.4.1 DAC: Shared Memory

For shared memory segment objects (henceforth SMSs), access checks are performed when the SMS is initially attached, and are not checked on each subsequent access. Changes to access controls (i.e., revocation) are effective with the next attempt to attach to the SMS (DA4.1).

In cases where an administrative user determines that immediate revocation of access to a SMS is required, the administrative user can reboot the computer, thus destroying the SMS and all access to it.

If a process requests deletion of a SMS, it is not deleted until the last process that is attached to the SMS detaches itself (or equivalently, the last process attached to the SMS terminates) (DA4.2).

The default access control on newly created SMSs is determined by the effective user ID and group ID of the process that created the SMS and the specific permissions requested by the process creating the SMS (DA4.3).

- The owning user and creating user of a newly created SMS will be the effective user ID of the creating process (DA4.4).
- The owning group and creating group of a newly created SMS will be the effective group ID of the creating process (DA4.5).
- The creating process must specify the initial access permissions on the SMS, or they are set to null and the object is inaccessible until the owner sets them (DA4.6).
- SMSs do not have ACLs as described above, they only have permission bits (DA4.7).

Access permissions can be changed by any process with an effective user ID equal to the owning user ID or creating user ID of the SMS (DA4.8). Access permissions can also be changed by any process with an effective user ID of 0, also known as running with the root identity (DA4.9).

6.2.4.4.2 DAC: Message Queues

For message queues, access checks are performed for each access request (e.g., to send or receive a message in the queue) (DA4.10). Changes to access controls (i.e., revocation) are effective upon the next request for access (DA4.11). That is, the change affects all future send and receive operations, except if a process has already made a request for the message queue and is waiting for its availability (e.g., a process is waiting to receive a message), in which case the access change is not effective for that process until the next request (DA4.12).

If a process requests deletion of a message queue, it is not deleted until the last process that is waiting for the message queue receives its message (or equivalently, the last process waiting for a message in the queue terminates) (DA4.13). However, once a message queue has been marked as deleted, additional processes cannot perform messaging operations and it cannot be undeleted (DA4.14).

The default access control on newly created message queues is determined by the effective user ID and group ID of the process that created the message queue and the specific permissions requested by the process creating the message queue.

- The owning user and creating user of a newly created message queue will be the effective user ID of the creating process.
- The owning group and creating group of a newly created message queue will be the effective group ID of the creating process.
- The initial access permissions on the message queue must be specified by the creating process, or they are set to null and the object is inaccessible until the owner sets them.
- Message queues do not use ACLs as described above, they only have permission bits. (DA4.15)

Access permissions can be changed by any process with an effective user ID equal to the owning user ID or creating user ID of the message queue. Access permissions can also be changed by any process with appropriate privileges (DA4.16).

6.2.4.4.3 DAC: Semaphores

For semaphores, access checks are performed for each access request (e.g., to lock or unlock the semaphore) (DA4.17). Changes to access controls (i.e., revocation) are effective upon the next request for access (DA4.18). That is, the change affects all future semaphore operations, except if a process has already made a request for the semaphore and is waiting for its availability, in which case the access change is not effective for that process until the next request (DA4.19).

In cases where an administrative user determines that immediate revocation of access to a semaphore is required, the administrative user can reboot the computer, thus destroying the semaphore and any processes waiting for it. This method is the described in the Evaluated Configuration Guide. Since a semaphore exists only within a single host in the

network, rebooting the particular host where the semaphores is present is sufficient to revoke all access to that semaphore.

If a process requests deletion of a semaphore, it is not deleted until the last process that is waiting for the semaphore obtains its lock (or equivalently, the last process waiting for the semaphore terminates) (DA4.20). However, once a semaphore has been marked as deleted, additional processes cannot perform semaphore operations and it cannot be undeleted (DA4.21).

The default access control on newly created semaphores is determined by the effective user ID and group ID of the process that created the semaphore and the specific permissions requested by the process creating the semaphore (DA4.22).

- The owning user and creating user of a newly created semaphore will be the effective user ID of the creating process.
- The owning group and creating group of a newly created semaphore will be the effective group ID of the creating process.
- The initial access permissions on the semaphore must be specified by the creating process, or they are set to null and the object is inaccessible until the owner sets them.
- Semaphores do not have ACLs as described above, they only have permission bits (DA4.23).

Access permissions can be changed by any process with an effective user ID equal to the owning user ID or creating user ID of the semaphore (DA4.24). Access permissions can also be changed by any process with appropriate privileges (DA4.25).

This function contributes to satisfy the security requirements FDP_ACC.1(1), FDP_ACF.1(1), FIA_USB.1, FMT_MSA.1(1), FMT_SMF.1, FMT_MSA.3(1) and FMT_REV.1(2).

6.2.5 Mandatory Access Control (MA) (LSPP/RBAC mode only)

Mandatory access control in the TOE is based on sensitivity labels, which are checked by the SELinux module. Although SELinux provides the ability to implement a wealth of other security policies, the TOE implements mandatory access controls based on the Bell/LaPadula model of labeled access controls.

A sensitivity label is a tag consisting of of a sensitivity (or a range of sensitivities) and a set of categories. It is part of the security context attached to every subject and object.

The MAC functions implemented by the TOE are as follows:

6.2.5.1 Sensitivity labels (MA.1)

All subjects an objects are assigned a sensitivity label, consisting of a hierarchical security level (or a range of security levels) and a set of categories (MA.1.1). Sensitivity labels are part of the security context and hence attached to all subjects and objects covered by the access control policies. The management of sensitivity labels can be performed for each subject and each object separately.

Note that in filesystems that do not support extended attributes, objects inherit the security context of their mount point. For example, the VFAT file system mounted under /boot/efi inherits the sensitivity label that was assigned to the mount point during the mount operation.

SELinux supports 16 hierarchical sensitivity levels (s0 to s15) and 1024 categories (c0 to c1023) (MA.1.2).

At login time, a user is assigned the default sensitivity label (the lower bound of the range of sensitivity labels) and the default set of categories associated with the account (MA.1.3). Users cannot change their sensitivity label or categories to values outside the range assigned to the account. (MA.1.4).

This function contributes to satisfy the security requirements FDP_IFC.1, FDP_IFF.2, FIA_ATD.1, FIA_USB.1, FMT_MSA.1(2), FMT_MSA.3(2).

6.2.5.2 MAC checks (MA.2)

The access check algorithm implemented by the Mandatory Access Control policy is as follows:

- When comparing two sensitivity labels A and B, sensitivity label A *dominates* sensitivity label B if label A is equal or greater than label B (as defined in FDP_IFF.2.7 a). For sentitivity labels of subjects with different lower and upper bounds (i.e. with a real "range"), the lower bound is used in the checks as the "effective" sensitivity level. If objects have a different lower and upper bound of a sensitivity label, the mechanism verifies whether the subjects "effective" sensitivity label is within the range of the object's label range.
- A subject can perform a read (or equivalent) operation on an object only if its sensitivity label dominates the object's sensitivity label. (MA.2.1)
- A subject can perform a write (or equivalent) operation on an object only if the object's sensitivity label is equal the subject's sensitivity label. (MA.2.2). Note that equality is still a domination; SELinux therefore further restricts the LSPP MAC policy. If users wish to write an object at a higher label, they can do so by transitioning to that label (using the newrole command), which effectively limits write-up to the upper bound of the user's label range.
- A subject in the secadm_r role allows the configuration of the SELinux mechanism as defined in section 6.1.2.3. This effectively implements a MAC override privilege (MA.2.3).

The revocation of access rights for a user is enforced upon the next access check for the MAC policy (MA.2.4).

This function contributes to satisfy the security requirement FAU_SAR.2, FDP_IFF.2, FMT_REV.1(2).

6.2.5.3 Import and export of labeled and unlabeled data (MA.3)

As the security context of a persistent object (including the sensitivity label) is stored in the file's extended attributes, sensitivity labels can be exported from and imported into the system together with the file they are attached to. RHEL provides the *star* command for export and import of files with their extended attributes. (MA.3.1).

Users can export data without labels to single-level devices allocated to them, if

1. the user's sensitivity label is in the set of the device's sensitivity label;
2. the device's sensitivity label equals the sensitivity label of the data written to it. (MA.3.2)

When transporting data over the network, the IPSec or CIPSO protocols can be used to preserve the sensitivity labels of the data in transmission (MA.3.3).

On object creation, the object's sensitivity label is set according to the transition rules of the policy, which use the subjects context and other security attributes (like the containing object's security context), depending on the object class of the newly created object. (MA.3.4).

Data can be exported without a label to a single-level device only (MA.3.5). The label of that device must equal the sensitivity label of the exported data. Upon import of unlabeled data, the label of the user causing the import is used for the imported data (MA.3.6).

This function contributes to satisfy the security requirements FDP_ETC.1, FDP_ETC.2, FDP_ITC.1, FDP_ITC.2 and FPT_TDC.1.

6.2.5.4 Printing labeled data (MA.4)

Printing of files is performed by utilities which enforce the file's sensitivity label to be printed together with the contents (MA.4.1).

It is possible to define multiple print queues and assign a single security level to each. This allows unlabeled printing (i.e., for PostScript documents) in a secure manner (MA.4.2).

For printing labeled data, the print spooler converts any input information into bitmaps and subsequently puts the applicable label information on a separate banner and trailer page and on the header and bottom of each page (MA.4.3). The conversion to a bitmap prior to adding the label information ensures that the labels are always on top of any user specified information. After the adding of the label information, the print spooler encapsulates the bitmaps into PCL or PostScript with the versions stated in section 2.4.2 (depending on the configuration) and forwards it to the printer. The

printer therefore must support these configured language and must be connected with the connections allowed for LSPP mode as outlined in section 2.4.2.

This function contributes to satisfy the security requirement FDP_ETC.2.

6.2.6 Role-based Access Control (RBAC) (LSPP/RBAC mode only)

6.2.6.1 Role definition and privileges (RBAC.1)

As explained in section 6.1.2, each subject has a security context, which contains a role and an SELinux user identitiy. The TOE is already preconfigured with different roles as defined in 6.1.2.3 (RBAC1.1).

The administrator defines the default role for each user. Only administrators can modify subject / role associations. Users can change to another role if they have the role in their set of allowed roles (RBAC1.2). Note that for the TOE, this only applies to users in the staff_r role; these users are allowed to transition to the secadm_r or sysadm_r role, if they hold this role in their set of allowed roles.

Every subject and object has a role assigned in their security context (RBAC1.3).

Every subject can hold only one role at any time as their active role (RBAC1.4).

This function contributes to satisfy the security requirements FDP_ACC.1(2), FDP.ACF.1(2), FIA_ATD.1, FIA_USB.1, FMT_MSA.1(3), FMT_MSA.3(3), FMT_MTD.1(5) and FMT_SMR.2.

6.2.6.2 Access control decisions (RBAC.2)

When checking an access request for RBAC, the SELinux Security Server uses

- the security context of the object,
- the security context of the subject
- the security class of the object

and looks up the security policy database to determine the set of allowed operations for the subject on the object. It returns three access vectors to the policy enforcement module:

- allowed – all allowed operations.
- auditallow – the set of operations that will generate a log entry even if the operation is allowed
- dontaudit – the set of operations that do not generate a log entry if the operation is denied.

The policy enforcement module allows an operation if the operation is allowed in the “allowed” access vector. (RBAC2.1)

Revoked access rights are enforced upon the next access check (RBAC.2.2).

The SELinux framework ensures that only valid SELinux labels are allowed. An SELinux label is only valid with regards to roles if a user-chosen role (e.g. newrole) is within the defined set of roles associated with the user.

For access control in RBAC, each user has one active role. A user may have more than one role he can use, but there is always only one active role. No user can have an empty set of roles. Associated with each role are certain domain types. The administrator configures which types belong to which roles and which roles a user can assume.

Access checks are done based on the current user's domain type, the accessed object's object type, and the access mode. Access rules are administrator-defined.

Changing to a new role is a privileged action which a user can do using the newrole program. That program checks that the user may actually change to that new role (the role must be assigned to that user by the administrator).

Administrative roles are defined in section 6.1.2.3 above.

This function contributes to satisfy the security requirement FAU_SAR.2, FDP_ACC.1(2), FDP.ACF.1(2), FMT_REV.1(2).

6.2.7 Object Reuse (OR)

Object Reuse is the mechanism that protects against scavenging, or being able to read information that is left over from a previous subject's actions. Explicit initialization is appropriate for most TSF-managed abstractions, where the resource is implemented by some TSF internal data structure whose contents are not visible outside the TSF: queues, datagrams, pipes, and devices. These resources are completely initialized when created, and have no information contents remaining.

Explicit clearing is used in Red Hat Enterprise Linux only for directory entries, because they are accessible in two ways: through TSF interfaces both for managing directories and for reading files. Because this exposes the internal structure of the resource, it must be explicitly cleared on release to prevent the internal state from remaining visible.

Storage management is used in conjunction with explicit initialization for object reuse on files, and processes. This technique keeps track of how storage is used, and whether it can safely be made available to a subject.

The following sections describe in detail how object reuse is handled for the different types of objects and data areas and how the requirements defined in FDP_RIP.2 are satisfied.

6.2.7.1 Object Reuse: File System Objects (OR.1)

All file system objects (FSOs) available to general users are accessed by a common mechanism for allocating disk storage and a common mechanism for paging data to and from disk. This includes the Journaling File System (ext3).

Object reuse is irrelevant for the CD-ROM File System (ISO-9660) because it is a read-only file system and so it is not possible for a user to read residual data left by a previous user. File systems on other media (tapes, diskettes.) are irrelevant because of warnings in the Evaluated Configuration Guide not to mount file systems on these devices. Also, object reuse for VFAT is irrelevant, since it is only used on a partition that is accessible exclusively by root.

Object reuse in the tmpfs file system is handled by the memory management object reuse functions. When allocating new space for a file, the TOE uses the functions of the memory management which clear the memory before it is allocated.

Object reuse for objects in the devpts file system is handled by the VFS layer. Note that devpts is not a disk based file system and therefore object reuse of disk space is not an issue for this file system.

The sysfs, binfmt_misc, and selinuxfs file systems correspond to a view of kernel parameters and are not disk based file systems, object reuse is not an issue for these file systems as the objects are under full control of the kernel and cannot be created or removed by users. The rootfs file system is only used temporarily during system startup and is completely inaccessible one startup is complete.

For this analysis, the term FSO refers not only to named file system objects (files, directories, device special files, named pipes, and UNIX domain sockets) but also to other abstractions that use file system storage (symbolic links and unnamed pipes). All of these, except unnamed pipes, have a directory entry that contains the last part of the pathname and an inode that controls access rights and points to the disk blocks used by the FSO.

In general, file system objects are created with no contents, directories and symbolic links are exceptions, and some of their content is specified at creation time (OR1.1).

This function contributes to satisfy the security requirement FDP_RIP.2.

6.2.7.2 Object Reuse: IPC Objects (OR.2)

Red Hat Enterprise Linux shared memory, message queues, and semaphores are initialized to all zeroes at creation. These objects are of a finite size (shared memory segment is from one byte to the value defined in /proc/sys/kernel/shmmax, semaphore is one bit), and so there is no way to grow the object beyond its initial size (OR2.1).

No processing is performed when the objects are accessed or when the objects are released back to the pool.

This function contributes to satisfy the security requirement FDP_RIP.2.

6.2.7.3 Object Reuse: Memory Objects (OR.3)

A new process's context is completely initialized from the process's parent when the fork system call is issued. All program visible aspects of the process context are fully initialized. All kernel data structures associated with the new process are copied from the parent process, then modified to describe the new process, and are fully initialized (OR3.1).

The Linux kernel zeroes each memory page before allocating it to a process. This pertains to memory in the program's data segment and memory in shared memory segments (OR3.2). When a process requests more memory from the kernel, the memory is explicitly cleared before the process can gain access to it (OR3.3). This does not include memory that has been buffered by the library routines used by process. But this memory has already been allocated to the process by the kernel (cleared for object reuse at that time). Note that process internal memory management and buffering is not subject of this Security Target.

When the kernel performs a context switch from one thread to another, it saves the previous thread's General Purpose Registers (GPRs) and restores the new thread's GPRs, completely overwriting any residual data left in the previous thread's registers (OR3.4). Floating Point Registers (FPRs) are saved only if a process has used them. The act of accessing an FPR causes the kernel to subsequently save and restore all the FPRs for the process, thus overwriting any residual data in those registers (OR3.5).

Processes are created with all attributes taken from the parent. The process inherits its memory (text and data segments), registers, and file descriptors from its parent (OR3.6). When a process execs a new program, the text segment is replaced entirely.

This function contributes to satisfy the security requirement FDP_RIP.2 and Note 1.

6.2.8 Security Management (SM)

This section describes the functions for the management of security attributes that exist within Red Hat Enterprise Linux.

6.2.8.1 Roles in CAPP mode (SM.1) (CAPP mode only)

A simple role model is used in the CAPP mode of operation that just supports two roles: administrative users and normal users (SM1.1).

In the evaluated configuration, a user has the role of an administrative user when he is allowed to *su* to root. The root account itself will not be used as a userid where a user can directly log in to (except for login from the system console). So every administrative user has his/her own userid, which is used to log into the system.

6.2.8.1.1 Administrative Users

Users that are allowed to *su* to an account with uid 0 can perform administrative actions in CAPP mode (provided they also know the password required to *su* to this account). Users that don't have the privilege to use *su* (i.e. are not a member of the "wheel" group) in their user profile can not perform administrative actions even if they know the root password (SM1.2).

In LSPP/RBAC mode, users need to be in the staff_r role and have the secadm_r, auditadm_r or sysadm_r role in their set of authorized roles to be able to change into the security administrator or system administrator role, which additionally restricts SM.1.2 (SM1.3).

6.2.8.1.2 Normal Users

Normal users can not perform actions that require privileges. They can only execute those SUID root programs they have access to or perform type transition allowed by the SELinux policy (SM1.4). In the evaluated configuration this is restricted to those programs they need such as the passwd program that allows a user to change his/her own password.

This function contributes to satisfy the security requirement FMT_SMR.2, FMT_MTD.1(3) and FMT_MTD.1(4).

6.2.8.2 Roles in LSPP/RBAC mode (SM.6) (LSPP/RBAC mode only)

Roles in SELinux are a means of grouping privileges that are required to perform administrative and security-related tasks.

In the LSPP/RBAC mode of operation, the system performs a mandatory access check after a successful DAC check to enforce (among other objectives) a role-based access control policy. As explained in section 6.1.2, each subject (i.e. process) has a security context, which includes a SELinux user identity and a role. A user logging into the system and thereby creating a subject acting on his behalf can use the *newrole* command to choose a role from the set of roles that have been assigned to his user account (SM6.1). Although a user can be assigned to more than one role, a subject is associated with only one role at any given time.

In the evaluated configuration, the system uses a set of fixed roles as described in 6.1.2.3 “SELinux roles”.

This function contributes to satisfy the security requirement FMT_SMR.2, FMT_MSA.1(2/3), FMT_MSA.3(2/3), FMT_MTD.1(5).

6.2.8.3 Access Control Configuration and Management (SM.2)

Access control to objects is defined by the permission bits or by the Access Control Lists (for those objects that have access control lists associated with them). Default access permission bits are defined in the system configuration files that define the value of the access control bits for objects being created without explicit definition of the permission bits. The administrative user can define and modify those default values.

Permissions can be changed by the object owner and an administrative user (SM2.1). When an object is created the creator is the object owner (SM2.2). Object ownership can be transferred (SM2.3). In the case of IPC objects, the creator will always have the same right as the owner, even when the ownership has been transferred (SM2.4).

This function contributes to satisfy the security requirements FMT_MSA.1(1), FMT_MSA.3(1), FMT_SMF.1 and FMT_REV.1(2).

6.2.8.4 Management of User, Group and Authentication Data (SM.3)

6.2.8.4.1 Creating new Users

An administrative user can create a new user and assigns a unique userid to this user. The initial password has to be defined using the *passwd* command. The new user will be disabled until the initial password is set (SM3.1).

Attributes that can be set for each user are among others (a complete list can be found in the description of the *useradd* command and the description of the content of the files */etc/passwd* and */etc/groups*):

- Administrative status of the user
- List of groups the user belongs to
- Home directory for this user

Those attributes are stored in the file */etc/passwd* and */etc/groups* (for the list of all groups the user belongs to). (SM3.2)

In LSPP/RBAC mode, the *semanage* command can be used by a user in the secadm_r role to set and modify the security context attributes (SELinux user, role, sensitivity label) of a user (SM.3.3).

6.2.8.4.2 Modification of user attributes

User attributes can be modified by an administrative user. Modifications of user attributes require the modification of the administration database that contains the user attributes (mainly */etc/passwd*) (SM3.4).

6.2.8.4.3 Management of Authentication Data

An administrative user has the capability to define rules and restrictions for passwords used to authenticate users. The parameters available are:

- The number of days (since January 1, 1970) since the password was last changed.

- The number of days before password may be changed (0 indicates it may be changed at any time)
- The number of days after which password must be changed (99999 indicates user can keep his or her password unchanged for many, many years)
- The number of days to warn user of an expiring password (7 for a full week)
- The number of days after password expires that account is disabled
(SM3.5)

All users are also allowed to change their own password using the *passwd* command. The password restrictions defined by the administrative user apply (SM3.6).

This list of attributes satisfies those required by FIA_ATD.1. In addition this function contributes to satisfy the security requirements FIA_SOS.1, FMT_MTD.1(3), FMT_MTD.1(4), FMT_SMF.1 and FMT_REV.1(1).

6.2.8.5 Management of Audit Configuration (SM.4)

The TOE allows configuring the events to be audited. Those events are defined in a specific configuration file and then the */etc/rc.d/init.d/auditd* script with the 'reload' parameter is used to notify the audit subsystem about modifications in the rules defining the events to be audited. The use of the */sbin/auditd* command and the */etc/rc.d/init.d/auditd* script is restricted to administrative users. In addition the TOE allows an administrative user to start or stop the audit subsystem (also using the */etc/rc.d/init.d/auditd* script to start the audit subsystem (using the 'start" parameter) or stop the audit subsystem (using the 'stop' parameter) (SM4.1).

The administrative user can define the events to be audited in form of a set of rules using simple filter expressions (SM4.2).

In LSPP/RBAC mode, the SELinux policy defines which operations will be audited if they succeed or fail, by providing the auditallow and dontaudit vectors when an RBAC access check is made (SM.4.3)

This function contributes to satisfy the security requirements FAU_GEN.1 and FAU_SEL.1 as well as FMT_MTD.1(1) and FMT_MTD.1(2)

6.2.8.6 Reliable Time Stamps (SM.5)

The TOE maintains a reliable clock used to generate time stamps as required for the TOE itself and applications. The audit subsystem requires such a reliable time source for the date and time field in the header of each audit record. The clock uses timers provided by the hardware and interrupt routines that update the value of the clock maintained by the TOE.

The initial value for this clock may be provided by a hardware clock that is part of the TOE hardware, by a trusted external time source (e. g. via the ntp protocol) or by the system administrator setting the initial value. Hardware time sources that are not found on the TOE hardware but are connected to the TOE hardware as auxiliary hardware are part of the TOE environment. Only the system administrator is allowed to overwrite the value of the clock maintained by the TOE (e. g. to correct the value in case it has drifted over time due to some inaccuracy of the hardware timer used by the TOE) (SM5.1).

This function contributes to satisfy the security requirement FPT_STM.1

6.2.8.7 SELinux Security Management (SM.7) (LSPP/RBAC mode only)

Management of the security policy and security contexts in SELinux is performed by the security administrator, i.e. a user in the secadm_r role: The different management activities are as follows:

- Assigning existing Linux users to SELinux user identities and roles (SM.7.1). SELinux provides the semanage command to accomplish this administration task.
- Installing a new SELinux policy and re-labeling objects according to this policy. Note that although the security administrator has the power to change the policy, any policy change would constitute a deviation

from the evaluated configuration. It is therefore not envisaged that security administrators change the policy other than for the initial configuration of the system. (SM.7.2)

- Changing the security context of individual objects. The security administrator can use the chcon and restorecon commands to set a new context for an object or to revert the object's context to the default context as stored in the SELinux policy database. (SM.7.3)

The auditadm_r role performs the following management functions:

- Managing the audit trail as described in section 6.2.3 "Audit (AU)" (SM.7.4).

This function contributes to satisfy the security requirement FMT_MTD.1(3), FMT_MTD.1(4), FMT_REV.1(1), FMT_REV.1(2)

6.2.9 Secure Communication (SC)

The TOE provides the ability to protect communication by cryptographic mechanism against disclosure and undetected unauthorized modification. The TOE supports two protocols (SSH v2 and SSL v3) that provide protection of communication against the above mentioned threats. **Note that communication using other protocols is not protected against those threats.**

The protocols SSH v2 and SSL v3 allow a secure communication between the TOE and a remote trusted IT product (which may be another instantiation of the TOE itself) over an insecure network. Within the TOE the protocols are configured to allow the secure tunneling of TCP based protocols. The difference between the two possibilities for tunneling consists in the authentication involved.

In the case of the SSH protocol the TOE supports establishing a secure connection allowing an application on a client system to set up the communication to the server side system after successful user authentication. This allows to get access to a shell from a remote system but also to perform actions such as secure file transfer where access to the files on the remote system is protected by the discretionary access control mechanism.

In the case of the SSL protocol, the TOE would allow to set up a secure communication channel between a client and an untrusted application (e. g. a web server) on the server side. This would allow a client to access the web server without user authentication but (depending on the configuration of the SSL server) with the certificate based authentication of the client system.

6.2.9.1 Secure Protocols (SC.1)

The TOE offers several protocols that applications can use to securely communicate with another trusted IT product (provided this supports those protocols in the same way as the TOE does):

- Secure Shell Protocol Version 2 (SSH v2) (SC.1.1)
- Secure Socket Layer Protocol Version 3 (SSL v3) (SC.1.2)

All protocols are able to establish a secure channel between a client and a server process. The TOE supports both the client as well as the server processes for all of those protocols and therefore is able to initiate a connection as well as act as the receiver part. All protocols provide the ability to "tunnel" an otherwise unprotected single port TCP based protocol.

For this evaluation, it was appropriate for the Security Target to claim compliance with the external standard for cipher suites listed in the different iterations of FCS_COP.1 for the definition of the encryption algorithm. There are many ways of determining compliance with a standard. The TOE has chosen to make a developer claim of compliance supported with a verification by an independent FIPS accredited lab. This means that there has been an independent verification by the independent lab consistent with the NIST cryptographic algorithm validation program that the implementation of the cryptographic algorithms actually meets the claimed standards. Additional verification of ciphers not covered by the cryptographic algorithm validation program was conducted by the FIPS accredited lab.

6.2.9.1.1 The Secure Shell Protocol

The TOE provides the Secure Shell Protocol Version 2 (SSH v2) to allow users from a remote host to establish a secure connection and perform a logon to the TOE.

The following table clarifies implementation details where the SSH standards left room for the implementation to decide on certain features.

Reference	Description	Implementation details
[SSH-TRANS] section 5	Compatibility With Old SSH Versions	The openssh implementation is capable of interoperating with clients and servers using the old 1.x protocol. That functionality is explicitly disabled in the evaluated configuration, it permits protocol version 2.0 exclusively.
[SSH-TRANS] section 6.2	Compression	openssh supports the OPTIONAL “zlib” compression method.
[SSH-TRANS] section 6.3	Encryption	The ciphers supported in the evaluated configuration are detailed below.
[SSH-AUTH] section 7	Public Key Authentication Method: “publickey”	This REQUIRED authentication method is supported by the openssh implementation but disabled in the evaluated configuration, it permits password authentication exclusively.
[SSH-AUTH] section 8	Password Authentication Method: “password”	This SHOULD authentication method is supported by openssl and is the only authentication method used in the evaluated configuration.
[SSH-AUTH] section 8	Password change request and setting new password	The openssl implementation does not support the optional password change mechanism in the evaluated configuration. Users must use the passwd(1) program (after successfully logging in) to do so.
[SSH-AUTH] section 9	Host-Based Authentication: “hostbased”	This OPTIONAL authentication method is disabled in the evaluated configuration.

The TOE supports the following security functions of the SSH v2 protocol:

1. Establishing a secure communication channel using the following cryptographic functions provided by the SSH v2 protocol:
    - Encryption using three key Triple DES in CBC mode (3des-cbc as defined in section 6.3 of [SSH-TRANS]) (SC1.3)
    - Diffie-Hellman key exchange (diffie-hellman-group1-sha1 as defined in section 6.5 of [SSH-TRANS]) (SC1.4)
    - The keyed hash function hmac-sha1 for integrity protection as defined in section 6.4 of [SSH-TRANS] (which refers to [RFC2104] for the exact definition of the algorithm) (SC1.5).

    Note: The protocol supports more cryptographic algorithms than the ones listed above. Those other algorithms are not covered by this evaluation and should be disabled or not used when running the evaluated configuration.
2. Performing user authentication using the standard password-based authentication method the TOE provides for users (SC1.6).

    Note: The protocol also supports other authentication methods (e. g. certificate based authentication) but those are not within the scope of this Security Target. This Security Target requires password based authentication and therefore the SSH v2 server should be configured to accept this authentication method only.
3. Checking the integrity of the messages exchanged and close down the connection in case an integrity error is detected (SC1.7).

6.2.9.1.2 The Secure Socket Layer Protocol

The TOE provides the Secure Socket Layer Protocol Version 3 (SSL v3) to allow users from a remote host to establish a secure channel to the TOE. In contrast to the Secure Shell protocol described above, the SSL protocol does not support user authentication as part of the protocol. The SSL protocol within the TOE also allows tunneling other TCP based protocols (that satisfy the restrictions defined in the Evaluated Configuration Guide) securely between a client and a server system.

On the client as well as on the server side the Stunnel program can be used to tunnel non-SSL aware daemons and protocols (such as POP, IMAP, LDAP, etc) by having Stunnel provide the encryption, requiring no changes to the daemon's code. Stunnel acts as a trusted wrapper that can be used by applications implementing otherwise non-secure protocols. Stunnel as part of the TSF will ensure that the user data transmitted by those applications over the network will be confidentiality and integrity protected by the SSL v3 protocol. For guidance on how to set up such trusted channel and how to use it by applications please see the Evaluated Configuration Guide.

The Stunnel daemon will be configured to support the following cypher suites defined in the SSL v3 protocol or RFC3268:

- SSL_RSA_WITH_RC4_128_SHA
- SSL_RSA_WITH_3DES_EDE_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA (SC1.8)

Other cypher suites as defined in the SSL v3 specification or RFC3268 are not supported in this Security Target and the TOE should be configured to not support other cipher suites.

This implies that the following cryptographic algorithms from the OpenSSL library are used:

1. The RSA algorithm with 1024 bit modulus length. RSA is used for the exchange of the session key and for server authentication.
2. RC4 with a key size of 128 bit (as one alternative for the symmetric encryption algorithm)
3. Triple DES with a key size of 168 bit
4. AES with a key size of 128 or 256 bit
5. SHA-1 (as the cryptographic hash function)

An implication of the use of this cipher suite and its algorithms is the authentication of the SSL server site using digital certificates.

Note: The function to generate the RSA key pair used by the server is part of the TSF, but the generation of the certificate of the public key is regarded as an aspect of the IT environment. A widely accepted Certification Authority might be used to generate this certificate (allowing a wide community trusting this CA to validate the certificate). In a closed community it might also be sufficient to have one server within the community to act as a CA. The OpenSSL library provides the functions to set up such a CA, but those functions are not subject of this Security Target.

The following table clarifies implementation details concerning the openssl implementation's compliance to the relevant standards. It addresses areas where the standards permit different implementation choices such as optional features.

Reference	Description	Implementation details
[SSLv3] 5.5	Handshake protocol overview: certificates	The evaluated configuration always uses server certificates. Use of client certificates is optional.
[SSLv3] D.1	Temporary RSA keys	Not applicable, the evaluated configuration does not limit the size of encryption keys to 512 bits.
[SSLv3] D.2	Random Number Generation and Seeding	openssl uses data from the */dev/urandom* device, a persistent entropy pool file, and volatile system statistics to seed the PRNG.

[SSLv3] D.3	Certificates and authentication	The evaluated configuration supports verification of certificate chains, the details are beyond the scope of this Security Target.
[SSLv3] D.4	CipherSuites	The ciphers supported in the evaluated configuration are listed above.
[SSLv3] D.5	FORTEZZA	The FORTEZZA hardware encryption system is not supported in the evaluated configuration.
[SSLv3] E.	Version 2.0 Backward Combatibility	The openssl implementation supports the backwards compatible protocol, but this is disabled in the evaluated configuration. It permits use of SSLv3 exclusively.
[TLS-AES]	CipherSuites	The ciphers supported in the evaluated configuration are listed above.

6.2.9.1.3 IPSec Security associations

The TOE can be configured to establish IPSec security associations at the IP layer in accordance with RFCs 2401 through 2406 and 2410, 3947 and 3948 as well as the key management RFCs 2407 trough 2409.

IPSec associations are able to associate sensitivity labels with the data transmitted (SC1.9). When a subject accesses an IPSec association endpoint, MAC checking ensures the enforcement of the MAC policy (SC1.10).

This function (SC1) contributes to satisfy the security requirements FCS_CKM.1 (1-3), FCS_CKM.2 (1-4), FCS_COP.1 (1-3), FDP_ETC.2, FDP_ITC.2, FDP_UCT.1, FDP_UIT.1, FMT_MSA.2, FPT_TDC.1 and FTP_ITC.1.

6.2.10 TSF Protection (TP)

While in operation, the kernel software and data are protected by the hardware memory protection mechanisms described in the high level design and the hardware reference manuals for the underlying hardware. The memory and process management components of the kernel ensure a user process cannot access kernel storage or storage belonging to other processes (TP1.1).

Non-kernel TSF software and data are protected by DAC, RBAC and process isolation mechanisms. In the evaluated configuration, the directories and files holding TSF data are owned as appropriate by either administrative users or pseudo users not mapping to a human user's account. All these objects and the directories in their paths are protected by restrictive DAC access rights. (TP1.2). In LSPP/RBAC mode, further access restrictions are imposed by the security contexts associated with these objects. (TP1.3). Note that MAC only marginally contributes to the protection of system objects, as such files are usually labeled at "syslow".

The TSF and the hardware and firmware components are required to be physically protected from unauthorized access. The system kernel mediates all access to the hardware mechanisms themselves, other than program visible CPU instruction functions and main storage defined by the kernel to be directly accessible by a user process.

The boot image for each host with the evaluated TOE in the networked system is adequately protected.

6.2.10.1 TSF Invocation Guarantees (TP.1)

All system protected resources are managed by the TSF. Because all TSF data structures are protected, these resources can be directly manipulated only by the TSF, through defined TSF interfaces. This satisfies the condition that the TSF must be "always invoked" to manipulate protected resources (TP1.4).

Resources managed by the kernel software can only be manipulated while running in kernel mode (TP1.5).

Processes run in user mode and can call functions of the kernel only as the result of an exception or interrupt (TP1.6). The hardware and the kernel software handling these events and ensure that the kernel is entered only at pre-determined locations, and within pre-determined parameters. All kernel managed resources are protected such that only the kernel software is able to manipulate them.

Trusted processes implement resources managed outside the kernel. The trusted processes and the data defining the resources are protected as described above depending on the type of interface. For directly invoked trusted processes the program invocation mechanism ensures that the trusted process always starts in a protected environment at a predetermined point (TP1.7). Other trusted process interfaces are started during system initialization and use well defined protocol or file system mechanisms to receive requests (TP1.8).

Some system calls or parameter of system calls are reserved for trusted processes. When called the kernel checks that the calling process runs with an effective userid of 0 (TP1.9) and (in LSPP/RBAC mode) with the appropriate security context (TP1.10).

This function contributes to satisfy the security requirement FPT_RVM.1.

6.2.10.2 Kernel (TP.2)

The Red Hat Enterprise Linux software consists of a privileged kernel and a variety of non-kernel components (trusted processes). The kernel operates on behalf of all processes (subjects).

The kernel runs in the CPU's privileged mode and has access to all system memory. All kernel software, including kernel extensions and kernel processes, execute with kernel privileges but only defined subsystems within the kernel are part of the TSF. The kernel is entered by some event that causes a context switch such as a system call, I/O interrupt, or a program exception condition.

Upon entry the kernel determines the function to be performed, performs it, and, when finished, performs another context switch to return to user processing (eventually on behalf of a different subject) (TP2.1).

The kernel is shared by all processes, and manages system wide shared resources. It presents the primary programming interface for Red Hat Enterprise Linux in the form of system calls.

Because the kernel is shared among all processes, any process running "in the kernel" (that is, running in privileged hardware state as the result of a context switch) is able to directly reference the data structures that implement shared resources.

The major components of the kernel are memory management, process management, the file system, the system call interface, and the device drivers.

This function contributes to satisfy the security requirement FPT_SEP.1.

6.2.10.3 Kernel Modules (TP.3)

Red Hat Enterprise Linux supports dynamically loadable kernel modules that are loaded automatically on demand. Kernel modules are actually a part of the kernel that is not resident but loaded as part of the kernel when needed (TP3.1). Whenever a program wants the kernel to use a feature that is only available as a loadable module, and if the kernel hasn't got the module installed yet, the kernel will take care of the situation and make the best of it (TP3.2).

This is what happens:

- The kernel notices that a feature is requested that is not resident in the kernel.
- The kernel uses modprobe to load a module that fits this symbolic description.
- modprobe looks into its internal "alias" translation table to see if there is a match. This table can be reconfigured and expanded by having "alias" lines in "/etc/modprobe.conf".
- modprobe is then asked to insert the module(s) that it has decided that the kernel needs. Every module will be configured according to the "options" lines in "/etc/modprobe.conf".
- modprobe exits and tells the kernel that the request succeeded (or failed...)
- The kernel uses the freshly installed feature just as if it had been configured into the kernel as a "resident" part.
  (TP3.3)

In the TOE Kernel modules will be not be automatically removed from the kernel when they have not been used for a period of time. Removing them from the kernel needs to be done explicitly.

This function contributes to satisfy the security requirement FPT_SEP.1.

6.2.10.4 Trusted Processes (TP.4)

Trusted processes in Red Hat Enterprise Linux are processes running in user mode but with additional privileges.

A trusted process is distinguished from other user processes by the ability to affect or enforce the security policy. Some trusted processes implement security policies directly (e.g., identification and authentication) but many are trusted simply because they operate in an environment that confers the ability to access TSF data (e.g., programs run by administrative users or during system initialization).

Trusted processes have additional kernel interfaces available for their use (depending on the privileges they hold), but are limited to kernel-provided mechanisms for communication and data sharing, such as files for data storage and pipes, sockets and signals for communication.

The major functions implemented with trusted processes include user login, identification and authentication, batch processing, some network operations, system initialization, and system administration.

The kernel will check for each system call that requires privileges if the process that issued the call has those privileges (TP4.1). If not, the kernel will refuse to perform the system call. The kernel will also check for each access to an object protected by the any of DAC, MAC, or RBAC mechanisms, if the process has the required access rights for the attempted type of access. Note that MAC and RBAC checks occur in LSPP/RBAC mode only.

Any program executed with privileges has the ability to perform the actions of a trusted process. It is therefore important that a site operating a Red Hat Enterprise Linux system strictly controls those programs and prohibits that those programs are modified or that programs from untrusted sources are executed with privileges (TP4.2).

Trusted processes are not part of the kernel and (except for those processes that perform system initialization and identification and authentication) are not part of the TSF itself.

In LSPP/RBAC mode, the access controls based on security contexts provide compartmentalization of trusted processes within their domains, so that trusted processes in one domain cannot interfere (directly or via accessible resources) with trusted processes in other domains.

Trusted processes provide a contribution to security management and identification and authentication. For identification and authentication they contribute to satisfy the security functional requirements FIA_UAU.2, FIA_UAU.7 and FIA_UID.2.

This function also contributes to FPT_SEP.1.

6.2.10.5 TSF Databases (TP.5)

Table 6-1 identifies the primary TSF databases used in Red Hat Enterprise Linux and their purpose. These are listed both as individual files (by pathname) or collections of files.

With the exception of databases listed with the User attribute (which indicates that a user can read, but not write, the file), all of these databases shall only be accessible to administrative users. None of these databases shall be modifiable by a user other than an administrative user.

Those databases are part of the file system and therefore the file system protection mechanisms of the TOE have to be used to protect those databases from unauthorized access. It is the task of the persons responsible for setting up and administrating the system to ensure that the access control features of the TOE are used throughout the lifetime of the system to protect those databases.

Each host system within the TOE maintains its own TSF database. Synchronizing those databases is not performed in the evaluated configuration. If such synchronization is required by an organization it is the responsibility of an administrative user of the TOE to achieve this either manually or with some automated assistance.Table 6-1: Administrative Databases.
This table lists other administrative files used to configure the TSF.

Database	Purpose
/etc/aide.conf	Defines the configuration of the integrity checker aide

Database	Purpose
/etc/aide.conf	Defines the configuration of the integrity checker aide
/etc/audit/audit.rules	defines filters for auditable event record generation
/etc/audit/auditd.conf	configuration settings for audit subsystem operation (such as audit trace file location and disk space thresholds)
/etc/cron.{weekly hourly daily monthly}/*	contains programs to be scheduled by the cron daemon on a weekly, hourly, daily or monthly schedule
/etc/cron.allow	File containing users allowed to use crontab
/etc/cron.d/*	contains programs to be scheduled by the cron daemon
/etc/cron.deny	File containing users not allowed to use crontab. Evaluated only if no /etc/cron.allow exists. If an empty /etc/cron.deny exists and no "allow" file exists, all users are allowed to use crontab.
/etc/crontab	commands to be scheduled by the cron daemon
/etc/cups/cupsd.conf	CUPS print system configuration
/etc/group	Stores group names, supplemental GIDs, and group members for all system groups.
/etc/gshadow	Stores group passwords and group administrator information
/etc/hosts	Contains hostnames and their address for hosts in the network. This file is used to resolve a hostname into an Internet address in the absence of a domain name server
/etc/inittab	Describes the process started by init program at different run levels
/etc/ld.so.conf	File containing a list of colon, space, tab, newline, or comma spearated directories in which to search for libraries for run-time link bindings
/etc/localtime	Defines the local time zone information used for date/time input and display
/etc/login.defs	Defines various configuration options for the login process.
/etc/modprobe.conf	Configuration file for modprobe. modprobe automatically loads or unloads a module while taking into account its dependencies.
/etc/netlabel.rules	Configuration file for loading netlabelctl rules during startup
/etc/pam.d/*	This directory contains the configuration of PAM. In it there is one configuration for each application that performs identification and authorization. Each of the configuration file contains the PAM modules that are to be used for this procedure.
/etc/passwd	Stores user names, user IDs, primary group ID, user real name, home directory, shell for all system users.
/etc/racoon/racoon.conf	Racoon key management daemon configuration files
/etc/rc.d/init.d/*	System startup scripts
/etc/rc.d/init.d/auditd	startup script for the audit system
/etc/securetty	Contains device names of tty lines on which root is allowed to login
/etc/security/opasswd	Contains the password history for check of reuse of old passwords
/etc/security/rbac-self-test.conf	Contains configuration information for RBAC self test utility
/etc/selinux/config	Wether SELinux is enabled and whether the mode is permissive or enforcing.

Database	Purpose
/etc/aide.conf	Defines the configuration of the integrity checker aide
/etc/selinux/mls/contexts/initrc_context	Holds security contexts for init scripts.
/etc/selinux/mls/contexts/securetty_types	Contains device names of tty lines on which label change is allowed within a session
/etc/selinux/mls/modules/active/booleans.local	Runtime policy configuration.
/etc/selinux/mls/policy/policy.20	Binary SELinux policy file
/etc/selinux/mls/setrans.conf	Defines names for system representation of MLS labels.
/etc/selinux/mls/seusers	SELinux users file
/etc/security/namespace.conf	Configuration of the location of polyinstantiated directories
/etc/selinux/semanage.conf	Configuration of libsemanage for manipulation of the policy.
/etc/shadow	Defines user passwords in one-way encrypted form, plus additional characteristics
/etc/ssh/sshd_config	Contains ssh configuration parameter for the ssh server
/etc/stunnel/stunnel.conf	Configuration file for stunnel service (location is configurable)
/etc/stunnel/stunnel.pem	File with certificate and private key for stunnel service (location is configurable)
/etc/sysconfig/*	Directory containing several configuration files for network services
/etc/sysctl.conf	Defines kernel parameters
/etc/vsftpd/ftpusers	contains users not allowed to remotely access the system using the FTP protocol
/etc/vsftpd/vsftpd.conf	Contains configuration parameter for the vsftp server
/etc/xinetd.conf /etc/xinetd.d/*	Xinetd configuration file
/var/lib/aide/aide.db.gz /var/lib/aide/aide.db.new.gz	Database with checksum and meta data information about files checked by aide
/var/log/tallylog	Stores time and date of failed login attempts for each user.
/var/log/lastlog	Stores time and date of last successful login attempt for each user.
/var/spool/cron/root	Crontab file for the root user

These tables are not functions but they are part of the management of the TSF. As such they contribute to the system management security functional requirements FMT_MSA.3(1,2,3) and FMT_MTD.1(3,4,5) as well as FMT_SMF.1.

6.2.10.6 Internal TOE Protection Mechanisms (TP.6)

All kernel software has access to all of memory, and the ability to execute all instructions. In general, however, only memory containing kernel data structures is manipulated by kernel software. Parameters are copied to and from process storage (i.e., that accessible outside the kernel) by explicit internal mechanisms, and those interfaces only refer to storage belonging to the process that invoked the kernel (e.g., by a system call). Functions implemented in trusted processes are more strongly isolated than the kernel. Because there is no explicit sharing of data, as there is in the kernel address space, all communications and interactions between trusted processes take place explicitly through files and similar mechanisms.

This encourages an architecture in which specific TSF functions are implemented by well-defined groups of processes.

This function contributes to satisfy the security requirement FPT_SEP.1.

6.2.10.7 Testing the TOE Protection Mechanisms (TP.7)

The TOE provides a tool for administrative users that allow them to test the correct functions of the protection features of the underlying abstract machine. This tool performs tests on

1. the main memory (to check for failures in the memory hardware) (TP7.1)
2. the processor (to check the functions of the memory management unit and the separation between user and kernel mode) (TP7.2)
3. I/O devices (to check for correct operation of some I/O devices including the hard disks and the firmware used to access the disks) (TP7.3)

The tool generates a report on the tests performed and the results that those test had. The report is generated in human readable format and may be stored in a file or directed to a printer (TP7.4).

This function contributes to satisfy the security requirement FPT_AMT.1.

6.2.10.8 Testing the TSF (TP.8)

The TOE provides the tool aide to verify the correct operation of the TSF. This tool can be run by administrative users. The tool test:

- The integrity of the delivered executables and the correct settings of their DAC access rights; (TP.8.1)
- the conformance of object SELinux security contexts with the defined policy; (TP.8.2)

This function contributes to satisfy the security requirement FPT_TST.1.

6.2.10.9 Secure failure state (TP.9)

If the entire RBAC policy database or parts of it get corrupted or become inaccessible, all role-based access checks fail and access is denied (TP.9.1). After a failure or service discontinuity, the TSF enter a maintenance mode where the ability to return the TOE to a secure state is provided (TP.9.2). The TSF ensure that all checks work properly. If the check whether a privilege is associated with a role or the check whether a user has a role fail, the TSF will recover to a consistent and secure state (TP.9.3). The TSF ensure that the TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed (TP.9.4).

The TOE can experience a so called kernel panic. This can happen when, for example, an internal audit problem is detected, like running out of buffers before being able to flush kernel data to the audit subsystem (see the -f option in the auditclt man page). Once in kernel panic state, the system comes to a halt and doesn't allow user interaction. The machine has to be rebooted. On booting, a previous kernel panic is detected (because of certain files on the root file system), and the system stops in single user mode, allowing access only to an administrator. If the root file system is damaged beyond automatic repair, which would make it impossible to check for these special kernel panic files, the system will not boot either but go to single user mode and require an administrator to repair the file system first.

This function contributes to satisfy the security requirements FPT_FLS.1, FPT_RCV.1, and FPT_RCV.4.

6.3 Supporting functions part of the TSF

6.3.1 Processes executed by non-administrative users

The Red Hat Enterprise Linux TSF primarily exists to support the activities of user processes. A user, or non-TSF, process has no special privileges or security attributes. The user process is isolated from interference by other user processes primarily through the CPU execution state and address protection mechanisms and the way they are used by the kernel, and also through the protections on TSF interfaces for process and file manipulation.

User processes are by definition untrusted and therefore do not contribute to any security function. The TSF ensure that user processes are encapsulated in such a way that they are separated from the TSF and from processes (trusted and untrusted) running with different attributes and will only be able to communicate with them using the defined TSF interfaces. User processes therefore do not contribute to any security function of the TOE.

6.4 Assurance Measures

The following table provides an overview, how the assurance measures of EAL4 and ALC_FLR.3 are met by Red Hat Enterprise Linux.

Table 6-2: Mapping Assurance Requirements to Documentation

Assurance Component	Documentation describing how the requirements are met
ACM_AUT.1	The Red Hat Beehive configuration management system provides automation for the maintenance of the implementation representation as well as for the generation of the TOE.
ACM_CAP.4	Configuration management procedures within Red Hat are highly automated using a process supported by Configuration Management tools and the Red Hat build system.
ACM_SCP.2	Source code, generated binaries, documentation, test plan, test cases, test results, and security flaws are maintained under configuration management.
ADO_DEL.2	Red Hat Enterprise Linux is delivered on CD in shrink-wrapped package as well as via the Red Hat Network Internet delivery method to the customer. Red Hat verifies the integrity of the production CDs by checking a production sample. Since all packages are digitally signed the user is able and has to verify the integrity and authenticity of those packages.
ADO_IGS.1	Guidance for installation and system configuration is provided in the guidance documentation associated with the TOE.
ADV_FSP.2	The functional specification for Red Hat Enterprise Linux consists of the man pages that describe the system calls, the trusted commands as well as a description of the security relevant configuration files. A table with an explanation of the completeness of the definition of the TSFI provided by the sponsor lists all system calls, trusted commands and security relevant configuration files with a mapping to their description in the overall documentation.
ADV_HLD.2	A high level design of the security functions of Red Hat Enterprise Linux is provided. This document provides an overview of the implementation of the security functions within the subsystems of Red Hat Enterprise Linux and points to other existing documents for further details where appropriate.
ADV_IMP.1	As the TOE is Open Source, all source code is available to the evaluator.
ADV_LLD.1	The low-level design is provided to explain the security functionality of the Red Hat Enterprise Linux with a high level of detail. It is described in terms of modules.
ADV_RCR.1	The correspondence information is provided as part of the functional specification (with the spreadsheet). An additional document providing the correspondence to the TOE Summary Specification has been provided to the evaluation facility.
AGD_ADM.1	Red Hat provides a System Administration Guide, an Evaluated Configuration Guide and a Reference Guide as the main references for System Configuration and Administration. In addition, a guidance document outlining the evaluated configuration is provided.
AGD_USR.1	The Step-by-Step Guide, the Evaluated Configuration Guide and the Reference Guide contain the specifics for the secure usage of the evaluated configuration.
ALC_DVS.1	The Red Hat security procedures are defined and described in documents in the Red Hat intranet.
ALC_FLR.3	The defect handling procedure Red Hat has in place for the development of Red Hat Enterprise Linux requires the description of defects with their effects, security implications, fixes and required verification steps. The

Assurance Component	Documentation describing how the requirements are met
	process ensures a timely provision of the security fixes to customers.
ALC_LCD.1	Red Hat follows a life-cycle model to maintain the RHEL distribution.
ALC_TAT.1	All languages used for the implementation representation are fully defined with publicly well-known definitions as the TOE is an Open Source implementation.
ATE_COV.2	Detailed test plans are produced to test the functions of Red Hat Enterprise Linux. Those test plan include an analysis of the test coverage, an analysis of the functional interfaces tested and an analysis of the testing against the high level design.
ATE_DPT.1	Testing at internal interfaces is defined and described in the test plan documents and the test case descriptions
ATE_FUN.1	Testing has been performed on the platforms that are defined in the Security Target. Test results are documented such that the tests can be repeated.
ATE_IND.2	All the required resources to perform their own tests are provided to the evaluation facility to perform their test. The evaluation facility has performed and documented the tests they have created and performed as part of the evaluation technical report for testing.
AVA_MSU.2	A Misuse Analysis is provided by the sponsor.
AVA_SOF.1	The Strength of Function Analysis has been provided for the mechanism based on permutational or probabilistic algorithms as part of the developer's vulnerability analysis document.
AVA_VLA.2	A vulnerability analysis has been provided that describes the sponsor's approach to identify vulnerabilities of Red Hat Enterprise Linux as well as the results of the findings.

6.5 TOE Security Functions requiring a Strength of Function

The TOE has the password-based security function for identification and authentication (IA) that is implemented by a probabilistic or permutational mechanism. The mechanism rated in the strength of function analysis is the password mechanism for user authentication. The strength claimed for this function is SOF-medium. In addition the TOE uses cryptographic functions for the protection of communication links. The cryptographic algorithms used there are not subject to a strength of function analysis. Also the key generation process for the cryptographic algorithms supported by the TOE is not subject to a strength of function analysis.

7 Protection Profile Claims

7.1 PP Reference

This Security Target claims conformance with the CAPP, LSPP and RBAC protection profiles as defined in section 1.2 of this ST. See also the references section in the beginning of this ST.

CAPP and LSPP are listed on the TPEP web site of NSA as a "Certified Protection Profile".

7.2 PP Tailoring

7.2.1 Security Functional Requirements

The SFRs occurring in more than one of the protection profiles have been merged as follows:

- Common SFRs in LSPP and CAPP are identical, as LSPP is a superset of CAPP
- SFRs common to RBAC and LSPP/CAPP have been merged so that all functional requirements for that SFR stated in the protection profiles were included in the ST SFR, with the exceptions stated below.

There is one additional security functional requirement (FMT_SMF.1) that has been added to those defined in [CAPP], [LSPP] and [RBACPP]. The reason is that [CC] defines the new family FMT_SMF and adds dependencies from FMT_MSA.1 and FMT_MTD.1 to the new component FMT_SMF.1. To resolve those new dependencies, FMT_SMF.1 has been added as a security functional requirement in addition to those defined in the protection profiles.

FPT_TDC.1 has been included to satisfy a dependency to FDP_ITC.2, which has been forgotten in LSPP

Two SFRs (FIA_UAU.1 and FIA_UID.1) defined in the PP have been substituted by hierarchical superior ones (FIA_UAU.2 and FIA_UID.2). This does not affect the compliance to the Protection Profile. Since those components don't imply additional dependencies, the dependency analysis performed on the Protection Profile still applies.

Other requirements (FCS_CKM.1, FCS_CKM.2, FCS_COP.1, FDP_UCT.1, FDP_UIT.1, FMT_MSA.2 and FTP_ITC.1) represent TOE specific extensions to the requirements defined by [CAPP].

Security Functional Requirements have been refined where required by the Protection Profile.

One security functional requirement ("Note 1") is included in [CAPP] and [LSPP] as an extension to the requirements defined in part 2 of the Common Criteria. Aspects of conformance of structure and content of Note 1 with the Common Criteria requirements for extensions to part 2 are addressed in the evaluation of the Protection Profile. They are therefore not discussed in this Security Target.

In FMT.SMR.2.3, the [RBACPP] text put in parentheses in list item (b) makes an implementation assumption about administrative roles, stating that administrative roles automatically inherit the privileges of all object owners. This text has been removed, because this statement is not correct in the SELinux MLS environment, where sysadm_r inherits these privileges, but secadm_r does not. Note that this statement is wrong even for other RBAC implementations, because assigning all such privileges to *every* administrative role would defeat the very purpose of having different administrative roles.

The "inline iterations" for FMT_MSA.1 and FMT_MSA.3 found in LSPP have been changed to correct iterations FMT_MSA.1(1/2) and FMT_MSA.3(1/2), respectively.

7.2.2 Threats, Policies, Assumptions and Objectives

This ST combines the threats, policies, assumptions and objectives present in the different protection profiles and states them by their names as found in the respective protection profile.

The threats T.UAUSER, T.UAACCESS and T.COMPROT have been added to the threats defined by [RBACPP] (note that [CAPP] and [LSPP] only define OSPs).

One assumption on the TOE environment (A.NET_COMP) has been added to reflect the distributed nature of the TOE.

One security objective for the TOE (O.COMPROT) has been added to the objectives provided by the protection profiles to reflect the objective of being able to establish an Inter-TSF trusted channel between the TOE and another trusted IT product.

The following security objectives for the TOE environment have been added to the environment objectives provided by the protection profiles:

Table 7-1: Additional objectives for the TOE environment

OE.ADMIN	OE.INFO_PROTECT
OE.MAINTENANCE	OE.RECOVER
OE.SOFTWARE_IN	OE.SERIAL_LOGIN
OE.PROTECT	OE.CLASSIFICATION

OE.CLASSIFICATION has been introduced to address one of the many flaws in LSPP, which has no appropriate objective mapping to the assumptions A.CLEARANCE and A.SENSITIVITY.

Th other objectives are required to cover the specific threats addressing the TOE environment. All objectives are related to physical and procedural security measures and therefore address the TOE non-IT environment.

7.2.3 Assurance Requirements

The assurance requirements of the Protection Profile are those defined in the Evaluation Assurance Level EAL4 of the Common Criteria. This Security Target specifies an Evaluation Assurance Level EAL 4 augmented by ALC_FLR.3. Since the Evaluation Assurance Levels in the Common Criteria define a hierarchy, all assurance requirements of the Protection Profile are included in this Security Target. ALC_FLR.3 which has been added to the assurance requirements defined in the CAPP has no dependency on any other security functional requirement or security assurance requirement and is therefore an augmentation that has no effect on the security functional requirements or security assurance requirements stated in the Protection Profile.

8 Rationale

The rationale section provides additional information and demonstrates that the security objectives and the security functions defined in the previous chapter are consistent and sufficient to counter the threats defined in chapter 2.

In the different tables, entries that are relevant to the LSPP/RBAC mode of operation have been colored.

8.1 Security Objectives Rationale

The following tables provide a mapping of security objectives to the environment defined by the threats, policies and assumptions, illustrating that each security objective covers at least one threat, assumption or policy and that each threat, assumption or policy is covered by at least one security objective. For a mapping of the different objectives found in [CAPP], [LSPP] and [RBACPP] to the objectives used in this ST, please refer to Table 8-14.

8.1.1 Security Objectives Coverage

Table 8-1: Mapping Objectives to threats, assumptions and policies

Objective	Threat / Policy
O.AUTHORIZATION	T.UAUSER, P.AUTHORIZED_USERS
O.DISCRETIONARY_ACCESS	T.UAACCESS, P.NEED_TO_KNOW
O.MANDATORY_ACCESS	P.CLASSIFICATION
O.RESIDUAL_INFO	P.NEED_TO_KNOW, T.UAACCESS
O.MANAGE	P.AUTHORIZED_USERS, P.NEED_TO_KNOW, T.UAUSER, T.OPERATE
O.ENFORCEMENT	P.AUTHORIZED_USERS, P.NEED_TO_KNOW
O.AUDITING	P.ACCOUNTABILITY
O.COMPROT	T.COMPROT, P.NEED_TO_KNOW
O.DUTY	T.ROLEDEV
O.HIERARCHICAL	T.ROLEDEV
O.ROLE	T.ROLEDEV, P.ACCESS

Table 8-2: Mapping objectives for the environment to threats, assumptions and policies

Env. Objective	Threat / Assumption / Policy
OE.ADMIN	A.MANAGE, A.NO_EVIL_ADMIN
OE.CREDEN	A.COOP
OE.INSTALL	TE.COR_FILE, A.MANAGE, A.NO_EVIL_ADMIN, A.PEER, A.NET_COMP
OE.PHYSICAL	A.LOCATE, A.PROTECT, A.CONNECT
OE.INFO_PROTECT	TE.COR_FILE, A.PROTECT, A.UTRAIN, A.UTRUST, A.ACCESS, A.OWNER
OE.MAINTENANCE	TE.HWMF
OE.RECOVER	A.MANAGE, TE.HWMF, TE.COR_FILE
OE.SOFTWARE_IN	P.NEED_TO_KNOW
OE.SERIAL_LOGIN	A.CONNECT
OE.PROTECT	TE.COR_FILE, A.NET_COMP, A.CONNECT
OE.CLASSIFICATION	A.CLEARANCE, A.SENSITIVITY

Table 8-3: Mapping threats to objectives

Threat	Objective
T.UAUSER	O.AUTHORIZATION, O.MANAGE
T.UAACCESS	O.DISCRETIONARY_ACCESS, O.RESIDUAL_INFO
T.COMPROT	O.COMPROT
T.OPERATE	O.MANAGE
T.ROLEDEV	O.DUTY, O.HIERARCHICAL, O.ROLE
TE.HWMF	OE.MAINTENANCE, OE.RECOVER
TE.COR_FILE	OE.PROTECT, OE.INSTALL, OE.INFO_PROTECT, OE.RECOVER

Table 8-4: Mapping Assumptions to Objectives

Assumption	Objective
A.LOCATE	OE.PHYSICAL
A.PROTECT	OE.INFO_PROTECT, OE.PHYSICAL
A.MANAGE	OE.ADMIN, OE.INSTALL, OE.RECOVER
A.NO_EVIL_ADMIN	OE.ADMIN, OE.INSTALL
A.COOP	OE.CREDEN
A.UTRAIN	OE.INFO_PROTECT
A.UTRUST	OE.INFO_PROTECT
A.ACCESS	OE.INFO_PROTECT
A.OWNER	OE.INFO_PROTECT
A.CLEARANCE	OE.CLASSIFICATION
A.SENSITIVITY	OE.CLASSIFICATION
A.NET_COMP	OE.PROTECT, OE.INSTALL
A.PEER	OE.INSTALL
A.CONNECT	OE.SERIAL_LOGIN, OE.PROTECT, OE.PHYSICAL

Table 8-5: Mapping Policies to Objectives

Policy	Objective
P.AUTHORIZED_USERS	O.AUTHORIZATION, O.MANAGE, O.ENFORCEMENT
P.NEED_TO_KNOW	O.DISCRETIONARY_ACCESS, O.MANAGE, O.ENFORCEMENT, O.RESIDUAL_INFO, O.COMPROT, OE.SOFTWARE_IN
P.ACCOUNTABILITY	O.AUDITING
P.ACCESS	O.ROLE
P.CLASSIFICATION	O.MANDATORY_ACCESS

8.1.2 Security Objectives Sufficiency

T.UAUSER: The threat of impersonization of an authorized user by an attacker is sufficiently diminished by O.AUTHORIZATION requiring proper authorization of users gaining access to the TOE. O.MANAGE ensures that only administrative users (which are assumed to be trustworthy) have the ability to add new users or modify the attributes of users. Together those objectives ensure that no unauthorized user can impersonate as an authorized user.

T.UAACCESS: The threat of an authorized user of the TOE accessing information resources without the permission from the user responsible for the resource is removed by O.DISCRETIONARY_ACCESS requiring access control for resources and the ability for authorized users to specify the access to their resources. This ensures that a user can access

a resource only if the requested type of access has been granted by the user responsible for the management of access rights to the resource. In addition O.RESIDUAL_INFO ensures that an authorized user can not gain access to the information contained in a resource after the resource has been released to the system for reuse.

T.COMPROT: The threat of user data being compromised or modified without being detected is removed by O.COMPROT requiring the ability to set up an Inter-TSF trusted channel between the TOE and another trusted IT product that protects user data being transferred over this channel from disclosure and undetected modification.

T.OPERATE: The threat of compromising the IT assets under control of the TOE due to improper administration and usage is countered by O.MANAGE requiring mechanisms for the administrator and the TOE user to operate the TOE securely.

T.ROLEDEV: The threat of assigning roles to users in a way that undermines the security of the TOE is mitigated by O.DUTY providing the separation of duties capability, O.HIERARCHICAL providing hierarchical roles for easier role assignments and O.ROLE limiting the operation of interfaces provided by the TOE to users which are members of a role permitted for the requested operation.

TE.HWMF: The threat of losing data due to hardware malfunction is mitigated by OE.MAINTENANCE requiring the invocation of diagnostic tools during preventative maintenance periods. In addition OE.RECOVER requires the organizational procedures to be set up that are able to recover critical data and restart operation in a secure mode in the case such a hardware malfunction happens.

TE.COR_FILE: The threat of undetected loss of integrity of security enforcing or relevant files of the TOE is diminished by OE.INSTALL requiring procedures for secure distribution, installation and configuration of systems thereby ensuring that the system has a secure initial state with the required protection of such files, OE.PROTECT requiring protection of transferred data in the network the TOE is connected to and OE.INFO_PROTECT requiring procedures for the appropriate definition of access rights to protect those files when the system is up and running. OE.RECOVER ensures that the system is securely recovered, which includes the verification of the integrity of security enforcing or security relevant files as part of the recovery procedures.

A.ACCESS: The assumption that access to resources is based on roles where a role reflects the user’s responsibilities towards the resources as implemented by OE.ADMIN requiring that administrators are competent to perform administrative tasks.

A.CLEARANCE: The procedures about granting users access to specific security levels are provided with OE.CLASSIFICATION which requires the administrators to provide the necessary clearance to users and assign appropriate labels to resources.

A.LOCATE: The assumption on physical protection of the processing resources of the TOE is covered by OE.PHYSICAL requiring physical protection.

A.PROTECT: The assumption on physical protection of all hard- and software as well as the network and peripheral cabling is covered by the objectives OE.INFO_PROTECT demanding the approval of network and peripheral cabling and OE.PHYSICAL requiring physical protection.

Note: Physical protection of the network components and cabling is required by A.PROTECT which may seem to be redundant to A.CONNECT. But A.CONNECT also addresses protection against passive wiretapping, which may be done without having physical access to a hardware component.

A.MANAGE: The assumption on competent administrators is covered by OE.ADMIN requiring competent and trustworthy administrators and OE.INSTALL requiring procedures for secure distribution, installation and configuration of systems as well as OE.RECOVER requiring the administrator to perform all the required actions to bring the TOE into a secure state after a system failure or discontinuity.

A.NO_EVIL_ADMIN: The assumption on administrators that are neither careless nor willfully negligent or hostile is covered by OE.ADMIN requiring competent and trustworthy administrators and OE.INSTALL requiring procedures for secure distribution, installation and configuration of systems.

A.COOP: The assumption on authorized users to act in a cooperating manner is covered by the objective OE.CREDEN requiring the safe storage and non-disclosure of authentication credentials.

A.NET_COMP: The assumption on network components to not modify transmitted data is covered by the objective OE.PROTECT requiring procedures and/or mechanisms to ensure a safe data transfer between systems as well as

OE.INSTALL requiring proper installation and configuration of all parts of the networked system thus including also components that are not part of the TOE.

A.OWNER: The limitation of creation and management of new data object is implemented by O.INFO_PROTECT requiring that users are trained for their respective tasks that do not pass information to other users without the appropriate right to access the information.

A.PEER: The assumption on the same management control and security policy constraints for systems with which the TOE communicates is covered by OE.INSTALL requiring procedures for secure distribution, installation and configuration of the networked system.

A.SENSITIVITY: The procedures about assigning sensitivity labels to resources and the output of resources are provided with OE.CLASSIFICATION which requires the administrators to provide the necessary clearance to users and assign appropriate labels to resources.

A.CONNECT: The assumption on controlled access to peripheral devices and protected internal communication paths is covered by OE.SERIAL_LOGIN for the protection of attached serial login devices, OE.PROTECT for the protection of data transferred between servers and OE.PHYSICAL requiring physical protection.

A.UTRAIN: The assumption on trained authorized users is covered by OE.INFO_PROTECT which requires that authorized users are trained to protect the data belonging to them.

A.UTRUST: The assumption on authorized users to be trusted to protect data is covered by OE.INFO_PROTECT which requires that authorized users are trusted to use the protection mechanisms of the TOE adequately to protect their data.

P.ACCESS: The access right to a particular resource is governed by the role of the requestee as defined by O.ROLE.

P.AUTHORIZED_USERS: The policy demanding that users have to be authorized for access to the system is implemented by O.AUTHORIZATION and supported by O.MANAGE allowing the management of this functions and O.ENFORCEMENT ensuring the correct invocation of the functions.

P.CLASSIFICATION: Access to a resource is limited based on the sensitivity label of the requestee as implemented by O.MANDATORY_ACCESS.

P.NEED_TO_KNOW: The policy to restrict access to and modification of information to authorized users which have a „need to know" for that information is implemented by O.DISCRETIONARY_ACCESS demanding an appropriate access control function that allows to define access rights down to the granularity of an individual user and O.COMPROT protecting user data during transmission to another trusted IT product.. It is supported by O.RESIDUAL_INFO ensuring that resources do not release such information during reuse and by OE.SOFTWARE_IN preventing users other than administrative users from installing new software that might affect the access control functionality. O.MANAGE allows administrative and normal users (for the files they own) to manage these functions, O. ENFORCEMENT ensures that the functions are invoked and operate correctly.

P.ACCOUNTABILITY: The policy to provide a means to hold users accountable for their activities is implemented by O.AUDITING providing the TOE with such functionality.

8.2 Security Requirements Rationale

This section provides the rationale for the internal consistency and completeness of the security functional requirements defined in this Security Target.

8.2.1 Internal Consistency of Requirements

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

The functional components were selected from CC components defined in part 2 of the Common Criteria. Functional component FMT_SMF.1 (Specification of Management Functions) has been added in accordance with [CC]. The use of component refinement was accomplished in accordance with CC guidelines. Functional requirement "Note 1" has been taken from the Controlled Access Protection Profile [CAPP] and Labeled Security Protection Profile [LSPP] and the justification for this extension has been addressed in the evaluation of this protection profile.

Multiple instantiation of identical or hierarchically-related components was used to clearly state the required functionality that exists in this Security Target.

For internal consistency of the requirements we provide the following rationale:

Audit

The requirements for auditing have been completely derived from [CAPP]. The rationale for those requirements is:

FAU_GEN.1 defines the events that the TOE is required to be able to audit. Those events are related to the other security functional requirements showing which event contributes to make users accountable for their actions with respect to the requirement. FAU_GEN.2 requires that the events are associated with the identity of the user that caused the event. Of course this can only be done if the user is known (which may not be the case for failed login attempts).

FAU_SAR.1 ensures that authorized administrators are able to evaluate the audit records with is supported by FMT_SMR.2 by defining roles, while FAU_SAR.2 requires that no other users can read the audit records (since they may contain sensitive information). Taking into account that the amount of audit records gathered may be very large, FAU_SAR.3 requires that the TOE provides the ability to search the audit records for a set that satisfies defined attributes.

To avoid that always all possible audit records are generated (which would result in an unacceptable overhead to the system performance and might easily fill up the available disk space) the TOE is required in FAU_SEL.1 to provide the possibility to restrict the events to be audited based on a set of defined attributes.

Requirement FAU_STG.1 defines that audit records need to be protected from unauthorized deletion and modification to ensure their completeness and correctness. Requirement FAU_STG.3 addresses the aspect that the system detects a shortage in the disk space that can be used to store the audit trail. In this case the administrator is informed about the potential problem and can take the necessary precautions to avoid a critical situation.

FAU_STG.4 addresses the problem that the TOE might not be able to record further audit records (e. g. due to the shortage of some resources). Also in this case the TOE needs to ensure that such a situation can not be misused by a user to bypass the auditing of critical activities. Otherwise a user might deliberately bring the TOE into a situation where it is no longer able to audit critical events just to avoid that a critical action he performs is audited.

Management of audit is addressed by FMT_MTD.1 for both the audit trail and audited events and FMT_SMF.1.

Secure Communication

The TOE provides two protocols that allow applications or users to securely communicate with other trusted IT products (which may be other instantiations of the TOE) as defined by FCS_CKM.1(1-3), FCS_CKM.2(1-4) and FCS_COP.1(1-3). Those protocols use cryptographic functions to ensure the confidentiality and integrity of the user data during transmission as required by FDP_UCT.1 (confidentiality) and FDP_UIT.1 (integrity). The two protocols – although based on the same library of cryptographic functions – use different cryptographic algorithms to provide the required protection.

Both protocols provide the ability to establish an Inter-TSF trusted channel, as required by FTP_ITC.1.

The secure generation of cryptographic keys used for secure communications is addressed by FMT_MSA.2.

Discretionary Access Control

FDP_ACC.1(1) requires the existence of a Discretionary Access Control Policy for file system objects and Inter Process Communication objects. The rules of this policy are described in FDP_ACF.1(1). Management of access rights is defined in FMT_MSA.1(1) and FMT_REV.1(2) together with the default values defined in FMT_MSA.3(1). To be effective a discretionary access control mechanism requires user's to be properly identified and authenticated (as required by FIA_UID.2 and FIA_UAU.2), proper binding of subjects to users (as required by FIA_USB.1), reference mediation (as required by FPT_RVM.1) and domain separation (as required by FPT_SEP.1). The policy is also supported by the requirement for residual information protection (FDP_RIP.2) which prohibits that users access information they are not authorized to via residuals remaining in objects that the allocate. Management of access rights is also described in FMT_SMF.1. Using the DAC mechanism, the audit trail is protected against access by untrusted subjects as defined in FAU_SAR.2. Trusted processes are protected by DAC against unauthorized modification as defined in FPT_SEP.1.

Mandatory Access Control (LSPP/RBAC mode)

FDP_IFC.1 defines the mandatory access control for the named objects based on sensitivity labels. The applicable rule set which is implemented is defined in FDP_IFF.2. In LSPP/RBAC mode, the import and export of data with its sensitivity labels and data without its sensitivity labels is subject to the mandatory access control policy. This is defined with the requirements FDP_ITC.1 and FDP_ITC.2 (for import) and FDP_ETC.1 and FDP_ETC.2 (for export). Assigning of sensitivity labels to a user upon login is defined in FIA_USB.1 and user attributes are defined in FIA_ATD.1. The management of the mandatory access control mechanism is defined with FMT_MSA.1(2). Assignment of initial values for the sensitivity labels when creating a named object is defined in FMT_MSA.3(2). Revocation of user attributes is given in FMT_REV.1(2). The consistent interpretation of remotely communicated sensitivity labels is covered by FPT_TDC.1.

Using the MAC mechanism, the audit trail is protected against access by untrusted subjects as defined in FAU_SAR.2.

Role-based Access Control (LSPP/RBAC mode)

FDP_ACC.1(2) and FDP_ACF.1(2) define the role-based access control policy. Roles are defined in FMT_SMR.2. Management of object security attributes is defined by the instantiations FMT_MSA.1(3) and their initial values by FMT_MSA.3(3). Assigning roles to a user upon login is defined in FIA_USB.1 and the user attributes are defined in FIA_ATD.1. Revocation of attributes is defined in FMT_REV.1(2).

Using the RBAC mechanism, the audit trail is protected against access by untrusted subjects as defined in FAU_SAR.2.

Identification and Authentication

As stated above Identification and Authentication is required for a useful discretionary access control based on the identity of individual users. FIA_UAU.2 and FIA_UID.2 require that users are authenticated before they can perform any action on the TOE. FIA_SOS.1 in conjunction with FMT_MTD.3 and FMT_MSA.2 ensures that the mechanism used for authentication (passwords) has a minimum strength and FIA_UAU.7 provides some level of protection against simple spoofing in the TOE environment. The TOE users with their security attributes are defined in FIA_ATD.1, Since the TOE implements processes acting on behalf of the user FIA_USB.1 ensures that those processes act within the limits defined for the user they are acting for (unless they are trusted to perform activities beyond the rights of the user). The TOE ensures that appropriate controls are in place for session establishment initiated by users as defined by FTA_LSA.1 and FTA_TSE.1.

Object Reuse

As stated above object reuse (as required by FDP_RIP.2 and Note 1) is a supporting function that prohibits easy access to information via residuals left in objects when they are re-allocated to another subject or object. As this the function supports the intention of the discretionary access control policy.

Security Management

The functions defined so far require several management functions as defined by FMT_SMF.1.

The first one is the management of access rights (as defined by all iterations of FMT_MSA.1 and FMT_REV.1 “Revocation of Object Attributes”). In addition new objects require having default access rights which are required by all iterations of FMT_MSA.3.

The second one is the management of users, which is defined in FMT_MTD.1(3) “Management of User Attributes” and FMT_REV.1 “Revocation of User Attributes” Management of user attributes related to RBAC is handled in FMT_MTD.1(5). Since passwords are used for authentication the management of this authentication data is also required in FMT_MTD.1(4) “Management of Authentication Data” including FIA_ATD.1 for the maintenance of user data and FIA_SOS.1 for password quality checking. Management of the audit subsystem is expressed by the requirements for the management of the audit trail (FMT_MTD.1(1) “Management of the Audit Trail”) and the management of the audit events (FMT_MTD.1(2) “Management of the Audit Events”). Management of the audit events is supported by the ability to select the events to be audited (FAU_GEN.1, FAU_SEL.1). In addition the TOE supports several roles (administrative user and normal user in CAPP mode, system administrator, security administrator, staff, and normal users in LSPP/CAPP mode) which is expressed by FMT_SMR.2.

Security management also comprises the management of a reliable time stamps. Such time stamps are essential for correct time information within audit records. Times stamps are addressed by FPT_STM.1.

TSF Protection

The TOE needs to ensure that users are limited in their activities by the boundaries defined by the access control policy. To ensure this the TSF need to check all access of users to protected objects (as required by FPT_RVM.1) and maintain

a domain for its own execution that protects it from inference and tampering by any subject that is not part of the TSF. This is expressed with the requirement FPT_SEP.1.

Trusted applications provide the identification and authentication interface for the user as defined in FIA_UAU.2, FIA_UID.2 and FIA_UAU.7. These databases are managed by trusted users as defined in FMT_SMF.1

The databases for user management are protected by the TSF as defined in FMT_MSA.3(1-3), FMT_MTD.1(3) and FMT_MTD.1(4).

The TOE also needs to provide a tool that allows the administrator to check the integrity of the underlying hardware and the correct operation of the TSF. Such ability is addressed by FPT_AMT.1 and FPT_TST.1.

The TOE will enter a secure state when critical security functions fail, and allow the administrator to perform repairs and re-enter the normal operating mode as set forth by FPT_FLS.1, FPT_RCV.1, and FPT_RCV.4. The TOE will switch to maintenance mode if security functionality cannot be enforced any more due to a failure [FPT_FLS.1].

The following table shows how the security functional requirements map to the objectives defined for the TOE.

Table 8-6: Mapping Objectives to Security Functional Requirements

Objective	Security Functional Requirement
O.AUTHORIZATION	User Attribute Definition (FIA_ATD.1) Strength of Authentication Data (FIA_SOS.1) Authentication (FIA_UAU.2) Protected Authentication Feedback (FIA_UAU.7) Identification (FIA_UID.2) User-Subject Binding (FIA_USB.1) Management of Authentication Data (FMT_MTD.1(4)) Secure Security Attributes (FMT_MSA.2) Secure TSF data (FMT_MTD.3) Limitation on Scope of Selectable Attributes (FTA_LSA.1) TOE session establishment (FTA_TSE.1)
O.DISCRETIONARY_ACCESS	Discretionary Access Control Policy (FDP_ACC.1(1)) Discretionary Access Control Functions (FDP_ACF.1(1)) User Attribute Definition (FIA_ATD.1) User-Subject Binding (FIA_USB.1) Management of Object Security Attributes (FMT_MSA.1(1)) Static Attribute Initialization (FMT_MSA.3(1)) Revocation of Object Attributes (FMT_REV.1)
O.RESIDUAL_INFO	Object Residual Information Protection (FDP_RIP.2) Subject Residual Information Protection (Note 1)
O.MANAGE	Selectable Audit Review (FAU_SAR.3) Management of Object Security Attributes (FMT_MSA.1(1-3)) Static Attribute Initialization (FMT_MSA.3(1-3)) Management of the Audit Trail (FMT_MTD.1(1)) Management of Audited Events (FMT_MTD.1(2)) Management of User Attributes (FMT_MTD.1(3)) Management of Authentication Data (FMT_MTD.1(4)) Management of RBAC TSF data (FMT_MTD.1(5)) Revocation of User Attributes (FMT_REV.1) Specification of Management Functions (FMT_SMF.1) Security Management Roles (FMT_SMR.2) Manual Recovery (FPT_RCV.1)

Objective	Security Functional Requirement
	Function Recovery (FPT_RCV.4)
O.ENFORCEMENT	Reference Mediation (FPT_RVM.1) Domain Separation (FPT_SEP.1) Abstract Machine Testing (FPT_AMT.1) Failure with preservation of secure state (FPT_FLS.1) TSF Self Test (FPT_TST.1)
O.AUDITING	Audit Data Generation (FAU_GEN.1) User Identity Association (FAU_GEN.2) Audit Review (FAU_SAR.1) Restricted Audit Review (FAU_SAR.2) Selectable Audit Review (FAU_SAR.3) Selective Audit (FAU_SEL.1) Guarantees of Audit Data Availability (FAU_STG.1) Action in Case of Possible Audit Data Loss (FAU_STG.3) Protection of Audit Data Loss (FAU_STG.4) Management of the Audit Trail (FMT_MTD.1(1)) Management of Audited Events (FMT_MTD.1(2)) Security Roles (FMT_SMR.2) Reliable Time Stamps (FPT_STM.1)
O.COMPROT	Cryptographic Key Generation (FCS_CKM.1 (1-3)) Cryptographic Key Distribution (FCS_CKM.2 (1-4)) Cryptographic Operation (FCS_COP.1 (1-3)) Basic data exchange confidentiality (FDP_UCT.1) Data Exchange Integrity (FDP_UIT.1) Secure Security Attributes (FMT_MSA.2) Inter-TSF Trusted Channel (FTP_ITC.1)
O.MANDATORY_ACCESS	Export of Unlabeled User Data (FDP_ETC.1) Export of Labeled User Data (FDP_ETC.2) Mandatory Access Control Policy (FDP_IFC.1) Mandatory Access Control Functions (FDP_IFF.1) Import of Unlabeled User Data (FDP_ITC.1) Import of Labeled User Data (FDP_ITC.2) User Attribute Definition (FIA_ATD.1) User-Subject Binding (FIA_USB.1) Revocation of Object Attributes (FMT_REV.1) Management of Object Security Attributes (FMT_MSA.1(2)) Static Attribute Initialization (FMT_MSA.3(2)) Inter-TST basic TSF data consistency (FPT_TDC.1)
O.DUTY	Security Roles (FMT_SMR.2)
O.HIERARCHICAL	Management of RBAC TSF data (FMT_MTD.1(5))
O.ROLE	Role-Based Access Control Policy (FDP_ACC.1(2)) Role-Based Access Control Functions (FDP_ACF.1(2)) User Attribute Definition (FIA_ATD.1) User-Subject Binding (FIA_USB.1) Management of Object Security Attributes (FMT_MSA.1(3)) Static Attribute Initialization (FMT_MSA.3(3)) Security Roles (FMT_SMR.2) Revocation of Object Attributes (FMT_REV.1)

O.AUTHORIZATION

The TSF must ensure that only authorized users gain access to the TOE and its resources. Users authorized to access the TOE have to use an identification and authentication process [FIA_UID.2, FIA_UAU.2]. To ensure authorized access to the TOE, authentication data is protected [FIA_ATD.1, FIA_UAU.7, FMT_MTD.1 "Management of Authentication Data"]. The strength of the authentication mechanism must be sufficient to ensure that unauthorized users can not easily impersonate an authorized user [FIA_SOS.1] and [FMT_MSA.2] supported by [FMT_MTD.3]. Proper authorization for subjects acting on behalf of users is also ensured [FIA_USB.1].

Limitations on establishing user sessions must be defined and enforced as set forth in [FTA_LSA.1, FTA_TSE.1].

O.DISCRETIONARY_ACCESS

The TSF must control access to resources based on identity of users. The TSF must allow authorized users to specify which resources may be accessed by which users.

Discretionary access control must have a defined scope of control [FDP_ACC.1(1)]. The rules of the DAC policy must be defined [FDP_ACF.1(1)]. 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 [FIA_ATD.1, FIA_USB.1]. Authorized users must be able to control who has access to objects [FMT_MSA.1(1)] and be able to revoke that access [FMT_REV.1 “Revocation of Object Attributes”]. Protection of named objects must be continuous, starting from object creation [FMT_MSA.3(1)].

O.AUDITING

The events to be audited must be defined [FAU_GEN.1], and must be associated with the identity of the user that caused the event [FAU_GEN.2]. An authorized administrator must be able to read the audit records [FAU_SAR.1], but other users must not be able to read audit information [FAU_SAR.2]. The administrative user must be able to search the audit events in the audit trail using defined criteria [FAU_SAR.3] and also must be able to define the events that are audited and the conditions under which they are audited [FAU_SEL.1]. All audit records must be provided with a reliable time stamp [FPT_STM.1]. The audit system must ensure that audit records are not deleted or modified [FAU_STG.1] and are not lost because of shortage of resources [FAU_STG.3 and FAU_STG.4]. The administrative user must be able to manage the audit trail [FMT_MTD.1 “Management of the audit trail”] and the audit events [FMT_MTD.1 “Management of the audit events”]. The maintenance of roles set forth in [FMT_SMR.2] allows that audit data can be mapped to security relevant actions.

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 objects in the TOE must be purged prior to the reuse of the object containing the residual information [FDP_RIP.2] and before a resource is given to a subject [Note 1].

O.MANAGE

The TSF must provide all the functions and facilities necessary to support the administrative users that are responsible for the management of TOE security.

Aspects that need to be managed must be defined [FMT_SMF.1]. The TSF must provide for an administrative user to manage the TOE [FMT_SMR.2]. The administrative user must be able to administer the audit subsystem and examine audit data [FMT_MTD.1 “Management of the Audit Trail” and FMT_MTD.1 “Management of the Audit Events”, and FAU_SAR.3 “Selectable Audit Review”], administer user accounts [FMT_MTD.1(3) “Management of User Attributes”, FMT_MTD.1(4) “Management of Authentication Data”, FMT_MTD.1(5) “Management of RBAC TSF data”, FMT_REV.1(2) “Revocation of User Attributes”], and administer object attributes [FMT_MSA.1(1-3)]. In addition the default values for access control need to be defined [FMT_MSA.3(1-3)].

O.ENFORCEMENT

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

The TSF must make and enforce the decisions of the TSP [FPT_RVM.1]. It must be protected from interference that would prevent it from performing its functions [FPT_SEP.1]. The correctness of this objective is further met through the assurance requirements defined in this Security Target.

The TSF must provide the administrator with tools that allow checking the integrity of the underlying hardware and the correct operation of the TSF [FPT_AMT.1] supported by FMT_TST.1. The TOE will switch to maintenance mode of a security functionality cannot be enforced any more due to a failure [FPT_FLS.1].

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.

O.COMPROT

The TSF must be able to establish an Inter-TSF trusted channel between itself and another trusted IT product [FTP_ITC.1] protecting the user data transferred from disclosure [FDP_UCT.1] and undetected modification [FDP_UIT.1]. This TSF uses cryptographic functions in the implementation that require securely generating keys [FCS_CKM.1], distributing keys [FCS_CKM.2] and performing the required cryptographic operations on the user data [FCS_COP.1]. Keys used must be secure enough such that they can not be guessed [FMT_MSA.2].

O.MANDATORY_ACCESS

The TSF implements the mandatory access control to resources based on the sensitivity labels of subjects and objects.

Rules for the import and export of labeled and unlabeled user data must be defined [FDP_ETC.1, FDP_ETC.2, FDP_ITC.1, FDP_ITC.2].

Mandatory access control is defined with a scope of control [FDP_IFC.1]. The rules of the MAC policy are defined by [FDP_IFF.1]. The security attributes of subjects applicable for the MAC policy are defined by [FIA_ATD.1, FIA_USB.1]. Authorized users manage the MAC policy according to [FMT_MSA.1(2)] and are able to revoke that access [FMT_REV.1(1) “Revocation of Object Attributes”]. Rules for the creation of objects are defined in [FMT_MSA.3(2)]. The consistent interpretation of remotely communicated sensitivity labels is covered by [FPT_TDC.1].

O.DUTY

The TOE provides the capability of enforcing ‘separation of duties’ as outlined by [FMT_SMR.2].

O.HIERARCHICAL

The TOE must provide the capability of defining hierarchical roles as required by [FMT_MTD.1(5)].

O.ROLE

The TOE must prevent users from gaining access to and performing operations on its resources/objects unless they have been granted access by the resource/object owner or they have been assigned to a role (by an authorized administrator) which permits those operations [FMT_SMR.2].

Role based access control must have a defined scope of control [FDP_ACC.1(2)]. The rules of the RBAC policy must be defined [FDP_ACF.1(2)]. The security attributes of objects used to enforce the RBAC policy must be defined. The security attributes of subjects used to enforce the RBAC policy must be defined [FIA_ATD.1, FIA_USB.1]. Authorized users must be able to control who has access to objects [FMT_MSA.1(3)] and be able to revoke that access [FMT_REV.1 “Revocation of Object Attributes”]. Protection of named objects must be continuous, starting from object creation [FMT_MSA.3(3)].

No security functions for the non-IT environment have been added, since the procedures that need to be implemented can (and probably will) be different for each site running the evaluated version of Red Hat Enterprise Linux. Therefore no specific security functional requirements and security functions for the non-IT environment have been defined in this Security Target. Individual sites running Red Hat Enterprise Linux should validate that the procedures and physical security measures they have put in place are sufficient to cover the security objectives defined for the environment of the TOE in this Security Target.

8.2.2 Security Requirements Coverage

The following table shows that each security functional requirement addresses at least one objective and provides a rationale why the objective is addressed by the requirement

Table 8-7: Mapping Security Functional Requirements to Objectives

SFR	Objectives
FAU_GEN.1	O.AUDITING
FAU_GEN.2	O.AUDITING
FAU_SAR.1	O.AUDITING
FAU_SAR.2	O.AUDITING
FAU_SAR.3	O.AUDITING, O.MANAGE
FAU_SEL.1	O.AUDITING
FAU_STG.1	O.AUDITING
FAU_STG.3	O.AUDITING
FAU_STG.4	O.AUDITING
FCS_CKM.1(1)	O.COMPROT
FCS_CKM.1(2)	O.COMPROT
FCS_CKM.1(3)	O.COMPROT
FCS_CKM.2(1)	O.COMPROT
FCS_CKM.2(2)	O.COMPROT
FCS_CKM.2(3)	O.COMPROT
FCS_CKM.2(4)	O.COMPROT
FCS_COP.1(1)	O.COMPROT
FCS_COP.1(2)	O.COMPROT
FCS_COP.1(3)	O.COMPROT
FDP_ACC.1(1)	O.DISCRETIONARY_ACCESS
FDP_ACC.1(2)	O.ROLE
FDP_ACF.1(1)	O.DISCRETIONARY_ACCESS
FDP_ACF.1(2)	O.ROLE
FDP_ETC.1	O.MANDATORY_ACCESS
FDP_ETC.2	O.MANDATORY_ACCESS
FDP_IFC.1	O.MANDATORY_ACCESS
FDP_IFF.2	O.MANDATORY_ACCESS
FDP_ITC.1	O.MANDATORY_ACCESS
FDP_ITC.2	O.MANDATORY_ACCESS
FDP_RIP.2	O.RESIDUAL_INFO
Note 1	O.RESIDUAL_INFO
FDP_UCT.1	O.COMPROT
FDP_UIT.1	O.COMPROT
FIA_ATD.1	O.AUTHORIZATION, O.DISCRETIONARY_ACCESS O.MANDATORY_ACCESS, O.ROLE
FIA_SOS.1	O.AUTHORIZATION
FIA_UAU.2	O.AUTHORIZATION
FIA_UAU.7	O.AUTHORIZATION
FIA_UID.2	O.AUTHORIZATION
FIA_USB.1	O.AUTHORIZATION, O.DISCRETIONARY_ACCESS O.MANDATORY_ACCESS, O.ROLE
FMT_MSA.1(1)	O.DISCRETIONARY_ACCESS,

SFR	Objectives
	O.MANAGE
FMT_MSA.1(2)	O.MANDATORY_ACCESS, O.MANAGE
FMT_MSA.1(3)	O.MANAGE, O.ROLE
FMT_MSA.2	O.COMPROT, O.AUTHORIZATION
FMT_MSA.3(1)	O.DISCRETIONARY_ACCESS, O.MANAGE
FMT_MSA.3(2)	O.MANDATORY_ACCESS, O.MANAGE
FMT_MSA.3(3)	O.MANAGE, O.ROLE
FMT_MTD.1(1) Audit Trail	O.AUDITING, O.MANAGE
FMT_MTD.1(2) Audited Events	O.AUDITING, O.MANAGE
FMT_MTD.1(3) User Attributes	O.MANAGE
FMT_MTD.1(4) Authentication Data	O.AUTHORIZATION, O.MANAGE
FMT_MTD.1(5)	O.HIERARCHICAL, O.MANAGE
FMT_MTD.3	O.AUTHORIZATION
FMT_REV.1(1) User Attributes	O.MANAGE
FMT_REV.1(2) Object Attributes	O.DISCRETIONARY_ACCESS, O.MANDATORY_ACCESS, O.ROLE
FMT_SMF.1	O.MANAGE
FMT_SMR.2	O.MANAGE, O.AUDITING, O.DUTY, O.ROLE
FPT_AMT.1	O.ENFORCEMENT
FPT_FLS.1	O.ENFORCEMENT
FPT_RCV.1	O.MANAGE
FPT_RCV.4	O.MANAGE
FPT_RVM.1	O.ENFORCEMENT
FPT_SEP.1	O.ENFORCEMENT
FPT_STM.1	O.AUDITING
FPT_TDC.1	O.MANDATORY_ACCESS
FPT_TST.1	O.ENFORCEMENT
FTA_LSA.1	O.AUTHORIZATION
FTA_TSE.1	O.AUTHORIZATION
FTP_ITC.1	O.COMPROT

8.2.3 Security Requirements Dependency Analysis

The following table shows the dependencies between the different security functional requirements and if they are resolved in this Security Target.

Table 8-8: Dependencies between Security Functional Requirements

Security Functional Requirement	Dependencies	Resolved
FAU_GEN.1	FPT_STM.1 Reliable time stamps	Yes
FAU_GEN.2	FAU_GEN.1 Audit data generation FIA_UID.1 Timing of identification	Yes
FAU_SAR.1	FAU_GEN.1 Audit data generation	Yes
FAU_SAR.2	FAU_SAR.1 Audit review	Yes
FAU_SAR.3	FAU_SAR.1 Audit review	Yes
FAU_SEL.1	FAU_GEN.1 Audit data generation FMT_MTD.1 Management of TSF data	Yes
FAU_STG.1	FAU_GEN.1 Audit data generation	Yes
FAU_STG.3	FAU_STG.1 Protected audit trail storage	Yes
FAU_STG.4	FAU_STG.1 Protected audit trail storage	Yes
FCS_CKM.1	[FCS_CKM.2 Cryptographic key distribution or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction FMT_MSA.2 Secure security attributes	No (see comment below)
FCS_CKM.2	[FDP_ITC.1 Import of user data without security attributes or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction FMT_MSA.2 Secure security attributes	No (see comment below)
FCS_COP.1	[FDP_ITC.1 Import of user data without security attributes or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction FMT_MSA.2 Secure security attributes	No (see comment below)
FDP_ACC.1	FDP_ACF.1 Security attribute based access control	Yes
FDP_ACF.1	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	Yes
FDP_ETC.1	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]	Yes
FDP_ETC.2	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]	Yes
FDP_IFC.1	FDP_IFF.1 Simple security attributes	Yes
FDP_IFF.2	FDP_IFC.1 Subset information flow control FMT_MSA.3 Static attribute initialisation	Yes
FDP_ITC.1	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_MSA.3 Static attribute initialisation	Yes
FDP_ITC.2	[FDP_ACC.1 Subset access control, or	Yes

Security Functional Requirement	Dependencies	Resolved
	FDP_IFC.1 Subset information flow control] [FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path]	
FDP_RIP.2	No dependencies.	Yes
Note 1	No dependencies	Yes
FDP_UCT.1	[FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path] [FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]	yes (FTP_ITC.1 and FDP_ACC.1)
FDP_UIT.1	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] [FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path]	yes (FTP_ITC.1 and FDP_ACC.1)
FIA_ATD.1	No dependencies	Yes
FIA_SOS.1	No dependencies	Yes
FIA_UAU.2	FIA_UID.1 Timing of identification	Yes
FIA_UAU.7	FIA_UAU.1 Timing of authentication	Yes
FIA_UID.2	No dependencies	Yes
FIA_USB.1	FIA_ATD.1 User attribute definition	Yes
FMT_MSA.1	[FDP_ACC.1 Subset access control or FDP_IFC.1 Subset information flow control] FMT_SMR.2 Security roles FMT_SMF.1Specification of management function	Yes
FMT_MSA.2	ADV_SPM.1 Security Policy Model [FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_MSA.1 Management of security attributes FMT_SMR.2 Security roles	Yes
FMT_MSA.3	FMT_MSA.1 Management of security attributes FMT_SMR.2 Security roles FMT_SMF.1 1Specification of management function	Yes
FMT_MTD.1	FMT_SMR.2 Security Roles FMT_SMF.1 Specification of management function	Yes
FMT_MTD.3	ADV_SPM.1 Security Policy Model FMT_MTD.1 Management of TSF Data	Yes
FMT_REV.1	FMT_SMR.2 Security roles	Yes
FMT_SMF.1	No dependencies	Yes
FMT_SMR.2	FIA_UID.1 Timing of identification	Yes
FPT_AMT.1	No dependencies	Yes
FPT_FLS.1	ADV_SPM.1 Security Policy Model	Yes
FPT_RCV.1	ADV_SPM.1 Security Policy Model ADV_ADM.1 Administrator Guidance	Yes
FPT_RCV.4	ADV_SPM.1 Security Policy Model	Yes
FPT_RVM.1	No dependencies	Yes
FPT_SEP.1	No dependencies	Yes
FPT_STM.1	No dependencies	Yes
FPT_TDC.1	No dependencies	Yes

Security Functional Requirement	Dependencies	Resolved
FPT_TST.1	FPT_AMT.1 Abstract Machine Testing	Yes
FTA_LSA.1	No dependencies	Yes
FTA_TSE.1	No dependencies	Yes
FTP_ITC.1	No dependencies	Yes

Comment

The security functional requirements FCS_CKM.1, FCS_CKM.2 and FCS_COP.1 all have a dependency on FCS_CKM.4 (Cryptographic key destruction). The TOE does not explicitly implement a key destruction function.

Key destruction is performed implicitly for the symmetric session keys used by the Object Reuse function, which ensures that memory used to temporarily store the symmetric session key is cleared before it is assigned to another subject or object. This applies for both main memory as well as disk space (the session keys might be written to disk space as part of the paging function of the TOE. They are not stored in ordinary files).

With respect to the long-term public-private key pairs, the key destruction is performed by deleting the file containing the key. The Object Reuse function of the TOE ensures that the disk space previously allocated to the file storing those keys is cleared before it is assigned to another subject or object.

The other dependencies of those security functional requirements are satisfied. The TOE does not import keys but generates all keys themselves as expressed in the security functional requirement FCS_CKM.1

Remarks

The dependencies of FIA_UAU.2, FIA_UAU.7 and FMT_SMR.2 on FIA_UID.1 are resolved with the inclusion of FIA_UID.2 which is hierarchical to FIA_UID.1

The dependencies of FMT_MSA.1 and FMT_MSA.3 on FMT_SMF.1 were introduced by [CC] and have been considered here.

The multiple instantiations of FMT_MTD.1 and FMT_REV.1 have been included in this table, since a multiple instantiation of one security functional requirement may in some cases result in the requirement for multiple instantiations of depending requirements. This is not the case here, since they all rely on the same simple role model of the TOE.

This table shows that no unresolved dependencies exist between security functional requirements.

There are also no unresolved dependencies between security assurance requirements. This is because the evaluation assurance level EAL4 has been defined such that no unresolved dependencies exist. The additional assurance component ALC_FLR.3 has no dependencies and therefore there are no unresolved dependencies for assurance components.

8.2.4 Strength of function

This Security Target claims a SOF rating SOF-medium. This claim applies for FIA_SOS.1, whereby it is stated that a ‘one off’ probability of guessing the password in 1,000,000 is given. The SFR is in turn consistent with the security objectives. A claim of SOF-medium is also consistent with the assumption of a non-hostile user community and the assumption on physical protection which prohibits that well-skilled, hostile attackers get physical access to the TOE.

No strength of function analysis is performed for the cryptographic algorithms supported by the TOE as well as the process of the generation of the keys used by those cryptographic algorithms.

8.2.5 Evaluation Assurance Level

This security target claims EAL4 augmented with ALC_FLR.3, which is seen appropriate for a well-controlled, non-hostile environment.

8.3 TOE Summary Specification Rationale

8.3.1 Security Functions Justification

The following table shows that the IT security functions, as specified in the TOE summary specification, meet all security functional requirements for the TOE and work together to satisfy the TOE security functional requirements.

Table 8-9: Mapping Security Functional Requirements to Security Functions

SFR	Security Functions (TOE Summary Specification)
FAU_GEN.1	The audit events are generally defined in **AU** explaining, how the events are generated by the TOE. The System Administrator is able to define the events to be audited, which is described in **SM.**
FAU_GEN.2	The concept of a Login ID that is kept for a user after his initial login is explained in **AU**. This allows tracing events to the user that caused them even if the user changes his real and / or effective and filesystem user ID (e. g. with the su command or with the execution of a SUID program).
FAU_SAR.1	The ability of the authorized administrator to read the audit trail and to convert the audit records into human readable format is explained in **AU**.
FAU_SAR.2	The ability to restrict access to the audit trail to authorized users is addressed in **AU** and enforcement is realized by **DA, MA, RBAC**.
FAU_SAR.3	The ability of the authorized administrator to search the audit trail for events matching defined search criteria is expressed in **AU**.
FAU_SEL.1	The ability of the authorized administrator to define the events to be audited using predicates and logical expressions is described in **AU** and **SM**.
FAU_STG.1	The use of the TOE's discretionary access control policy to protect the audit trail and the audit configuration files from access by anybody else than an authorized administrator is defined in **AU**.
FAU_STG.3	The ability to generate a syslog message when the disk space for auditing gets below a limit defined in the audit configuration file is described in **AU**.
FAU_STG.4	The ability to stop processes trying to generate audit records in case the audit trail is full is described in **AU**.
FCS_CKM.1(1-3)	The multiple instantiations of this security functional requirement are described in **SC** where the SSH v2, IPSec and SSL v3 protocols and the cipher suites supported by the evaluated configuration are defined together with the key generation functions used.
FCS_CKM.2(1-4)	The multiple instantiations of this security functional requirement are described in **SC** where the SSH v2, IPSec and SSL v3 protocols and the cipher suites supported by the evaluated configuration are defined together with the key exchange / key negotiation functions used.
FCS_COP.1(1-3)	The multiple instantiations of this security functional requirement are described in **SC** where the SSH v2, IPSec and SSL v3 protocols and the cipher suites supported by the evaluated configuration are defined with the cryptographic algorithms used by the cipher suites.
FDP_ACC.1(1)	The discretionary access control policy is based on **DA** defining permission bits for the subjects and objects as there are file system objects and IPC objects.
FDP_ACC.1(2) LSPP/RBAC mode	The role-base access control policy is described in **RBAC**, defining the security contexts for subjects and objects.
FDP_ACF.1(1)	The discretionary access control is realized as described by **DA**. There the individual mechanisms for access control depending on the object type are described in detail.
FDP_ACF.1(2) LSPP/RBAC mode	**RBAC** describes the RBAC access control decisions, using the security contexts of subjects and objects, applying the SELinux policies to them and providing a vector with allowed operations.

SFR	Security Functions (TOE Summary Specification)
FDP_ETC.1 LSPP/RBAC mode	**MA** describes export of unlabeled data by restricting such export to a single-level device. Such export can also be over a network, where **SC** provides the functionality to protect the transmission
FDP_ETC.2 LSPP/RBAC mode	**MA** describes export of labeled data with archiving utilites that preserve labels. When printed, **MA** ensures that labels are always printed with labeled data. This is also possible over network connections, where **SC** provides preservation of labels through IPSec.
FDP_IFC.1 LSPP/RBAC mode	The information flow policy based on sensitivity labels is implemented by **MA**, covering all subjects and objects.
FDP_IFF.2 LSPP/RBAC mode	The MAC functions described in **MA** implement the Bell/LaPadula model of access controls based on sensitivity labels.
FDP_ITC.1 LSPP/RBAC mode	**MA** describes import of unlabeled data into the system, applying the importing subject's sensitivity label to the data.
FDP_ITC.2 LSPP/RBAC mode	**MA** describes how appropriately authorized subjects to import multi-level data with utilities that preserve restore sensitivity labels with the security contexts. For import of data over network connections, **SC** provides IPSec, which preserves labels, but restricts data to be single-level per connection.
FDP_RIP.2	Object residual information protection is realized by security functions for object reuse (**OR**) on file system objects, IPC objects, queuing system objects and miscellaneous objects.
Note 1	The object reuse performed before an object is re-assigned to another subject are described in **OR**.
FDP_UCT.1	The description how the confidentiality of user data is protected when using the SSH v2 or SSL v3 protocol is described in **SC**.
FDP_UIT.1	The description how the user data is protected from unauthorized modifications and insertions when using the SSH v2 or SSL v3 protocol is described in **SC**.
FIA_ATD.1	Security attributes belonging to individual users are realized by the user I&A data management of **IA**. Management of user attributes is described in **SM**. The user attributes for the MAC and RBAC attributes are described in **MA** and **RBAC**, respectively.
FIA_SOS.1	The passwd function of **IA** is able to enforce the verification of secrets as required. System management commands can be used to define parameters that can be used to (hopefully) enhance the strength of the passwords chosen by the user. Password management including the possible parameter to enhance the strength of passwords are explained in **SM**.
FIA_UAU.2	Authentication of each user before any action is realized by **IA** (common authentication mechanism and interactive login and related mechanisms). Authentication is initiated by a trusted process. Trusted processes are described in **TP**.
FIA_UAU.7	The login mechanisms of **IA** provide only obscured feedback during authentication. Authentication feedback is managed by a trusted process. Trusted processes are described in **TP**.
FIA_UID.2	Identification of each user before any action is realized together with authentication as in **IA** (see above). Identification is initiated by a trusted process. Trusted processes are described in **TP**.
FIA_USB.1	The required binding between subjects and users is implemented by the su functionality of **IA** and login processing. There also the logoff process is described which releases the binding between subjects and users. The limitations imposed on the binding of DAC, MAC and RBAC security attributes (i.e. user ID, sensitivity label ranges and security context ranges) are described in **DA, MA** and **RBAC**.
FMT_MSA.1(1)	The management of object security attributes is implemented by the access control configuration and management function **SM**, the objects are described in **DA** (file system objects and IPC objects).

SFR	Security Functions (TOE Summary Specification)
FMT_MSA.1(2) LSPP/RBAC mode	Management of security contexts (which include sensitivity labels) as described in **MA** is described in **SM** as being restricted to the security administrator role.
FMT_MSA.1(3) LSPP/RBAC mode	Management of security contexts and SELinux policies as described in **RBAC** is described in **SM** as being restricted to the security administrator role. Management itself is subject to controls described in **RBAC**.
FMT_MSA.2	The acceptance of only secure values is related to the use of secure cryptographic keys. The key generation aspects are discussed in **SC** for the different cryptographic algorithms used. The enforcement of good-quality passwords through the definition of a password policy is described in **IA**.
FMT_MSA.3(1)	Restrictive default values for security attributes are defined for the objects when they are created. Default values can be defined by an administrative user for all object types and by the user for file system objects created under his control. (see above, i.e. **SM** and **DA**). Some default values are defined in TSF databases as defined in **TP**.
FMT_MSA.3(2) LSPP/RBAC mode	Assignment of sensitivity labels to objects during creation is defined in **MA**, based on the SELinux policy and the creating subject's label and privileges. Changing labels is only possible with appropriate privileges, as is the change of the SELinux policy itself, as described in **SM.** The policy and user database are part of the TSF databases described in **TP**.
FMT_MSA.3(3) LSPP/RBAC mode	Security contexts for RBAC (including role assignments) are described in **RBAC** and **SM.** Users are restricted to one role at a time. Some default values are defined in TSF databases as defined in **TP**.
FMT_MTD.1(1) Audit Trail	The protection and management of the audit trail is described in **AU** as well as in **SM**. There tools available for converting the audit data to human readable format as well as the tool for searching the audit trail data are described.
FMT_MTD.1(2) Audited Events	The way an authorized administrator can select the events to be audited is defined in **SM**.
FMT_MTD.1(3) User Attributes	User security attributes are protected as required by the user identification and authentication data management **IA** and during the creation of new users in **SM**. User attributes are stored in TSF databases described in **TP**.
FMT_MTD.1(4) Authentication Data	Initialization of authentication data is restricted to administrative users during the creation of new users in **SM**. Authentication data (in encrypted form) and attributes are stored in TSF databases described in **TP**. Users are allowed to change their own authentication data within the limits defined by an administrative user. This is described in **SM**
FMT_MTD.1(5) LSPP/RBAC mode	Roles and role hierarchies are defined in the SELinux policy (**RBAC**). Management of role and policy definition and assignment of users to roles is restricted to the security administrator, as described in **SM.**
FMT_MTD.3 LSPP/RBAC mode	**IA** describes how the password complexity parameters are set to ensure sufficient password strength.
FMT_REV.1(1) User Attributes	The revocation of user security attributes as required in FMT_REV.1 is realized by the user management functions of **SM**.
FMT_REV.1(2) Object Attributes	Revocation of object security attributes is realized by the access control configuration and management function **SM**. Revocation is enforced during the next access check by the **DA**, **MA** and **RBAC** policy engines.
FMT_SMF.1	Management of security functions is addressed in the following security functions: Object security attributes management: **DA** (File system objects and IPC objects). In addition the following management functions are defined: Audit trail management: **AU** and **SM**. Audit event management: **AU** and **SM**.

SFR	Security Functions (TOE Summary Specification)
	Object attribute management: **SM** User attribute management**: SM** Authentication management: **SM** and **IA** In addition most of the management functions use the TSF databases (**TP**) to store management configurations.
FMT_SMR.2	The required roles are maintained within the security management of the roles in function **SM** and defined in **RBAC**. The enforcement of roles restrict access to the audit trail as defined in **AU**.
FPT_AMT.1	The ability of the authorized administrator to test the functions of the underlying abstract machine are described in **TP**.
FPT_FLS.1 LSPP/RBAC mode	Secure failure of the system by going down into a maintenance mode is described in **TP**, which covers the failure of the RBAC database (i.e. the SELinux policy).
FPT_RCV.1 LSPP/RBAC mode	Secure failure of the system by going down into a maintenance mode is described in **TP**.
FPT_RCV.4 LSPP/RBAC mode	Recovery from a failure by bringing the system back up from maintenance mode after repairing the system is described in **TP**.
FPT_RVM.1	The TSF invocation guarantee functionality **TP** ensure that TSP enforcement functions are always invoked before functions in the TSC are allowed to proceed.
FPT_SEP.1	The required domain separation for the TSF is realized by the kernel functionality itself, the kernel modules and trusted processes as described in **TP**, the discretionary access control mechanism described in **DA** and the internal TOE protection mechanisms described in **TP**.
FPT_STM.1	The function for the generation of a reliable time stamp is defined in **SM**.
FPT_TDC.1	Consistent interpretation of sensitivity labels is achieved through association of labels to IPSec connections (**SC**) and archiving utilities preserving the security attributes of objects (**MA**)
FPT_TST.1 LSPP/RBAC mode	The ability of the authorized administrator to test the security functions of the TOE are described in **TP**.
FTA_LSA.1 LSPP/RBAC mode	Restricitions on the selection of sensitivity labels and security contexts from the allowed range for the user during login and when transitioning to other accounts or roles is described in **IA**.
FTA_TSE.1 LSPP/RBAC mode	This requirement is trivially fulfilled, because every user always has a role assigned in LSPP/RBAC mode during login; empty role sets cannot happen (**IA**).
FTP_ITC.1	The function for setting up a trusted channel between the TOE and another trusted IT product using the SSH v2 or SSL v3 protocol is described in **SC.**

This table shows how the security functions work together to satisfy the security functional requirements.

All security functions work together in a mutually supportive way and do not contradict each other:

Identification and authentication (IA) is implemented by user names, authenticated a password mechanism and enforced whenever a user tries to establish a new session with the TOE. The I&A mechanisms use TSF databases as defined in **TP** to store the necessary identification and authentication data, as well as all security attributes for the users used to establish the session and start subjects acting on the user's behalf. These databases are protected by access rights of the **DA, MA** and **RBAC** security functions. **OR** ensures that no authentication data leaks to unauthorized users. I&A depends on proper user and user security attribute management, which is provided by **SM**, as well as the management of the password policy to ensure sufficient complex passwords. IA must not be circumvented, which is ensured by **IA** itself, which provides its functionality for all access paths of users, and by **TP,** which ensures that the security domain of the TOE cannot be tampered with. When authentication occurs over network connections, **SC** provides the required confidentiality of that data and ensures that no impostors can inject false information.

Auditing (AU) is invoked when security-relevant events occur. AU writes records of these events with sufficient information to provide traceability and accountability to an audit trail, which is implemented with file system objects.

Therefore, **DA**, **RBAC** and **MA** protect these files from tampering and from unauthorized access. **SM** functions provide the necessary management functionality, which allow to configure the auditable events and to manage the audit trail. Auditing can only support accountability if it is accompanied by a means to identify the users who triggered a security-relevant event. The necessary identification and authentication of users is provided by **IA**, as well as the binding of these users to the subjects invoking the events. Auditing relies on the **TP** functions to ensure that the mechanism is always invoked when necessary anfd cannot be circumvented.

Discretionary Access Control (DA) provides access controls based on access rights granted by object owners to users and groups of users. This security function requires a binding between users and subjects, as access controls are based on user identities rather than subject identities; this is provided by **IA**. The management of access rights provided by **SM** is required to tailor the policy to the requirements of the users. DA is only one of three access control policies of the system. The other policies RBAC and MA do not interfere with DA, because the TOE ensures through **TP** that all three policies are enforced and must succeed before access to a resource is granted. Since the policies address different security aspects, they do not interfere with each other, although it may happen that one policy would allow access to a resource, while another denies it. This is no contradiction and does not lead to inconsistent decisions. **OR** ensures that all access control policies cannot be circumvented by information leakage.

Mandatory Access Control (MA) also requires IA to provide reliable user identification and the user-subject binding provided by **IA**. Management of the MAC attributes for users and objects if provided by **SM**, which also provides support for the management of the overall SELinux policy. **TP** provides the invocation guarantees that the policy must rely on, as well as the guarantee that the policy enforcement cannot be tampered with in unauthorized ways. MAC requires labels to be attached to subjects and objects. When objects are transferred between systems, such labels may not be lost. This support is provided by **SC**.

Role-based Access Control (RBAC) has the same requirements as DA and MA on **IA** for reliable user identification and the user-subject binding, **SM** for the required management functions, and **TP** for support of the policy enforcement.

Object Reuse (OR) is not a security function actively used by users. It provides support to other security functions as described for the security functions in this section. It requires, however, that **TP** sufficiently separates subjects and enforces access mechanisms, so that object reuse for a resource can work when cleaning of previous contents takes place during reallocation of the resource rather than on deallocation. Through the TP protection, other security functions cannot interfere with OR.

Secure Communications (SC) stores its configuration and keys in file system objects. As such it depends on sufficient protection of these files, which is provided by DA, MA and RBAC. The configuration and key files are part of the TSF database provided by **TP**, and requires protection from tampering and bypassing its functions by **TP**.

TSF Protection (TP) is the basis almost all other security functions are built on. As such, it does not require any of the other security functions to perform its duty. However, the sandboxes built by the type enforcement of SE Linux as part of **RBAC** help in TP's separation task and enhance the TOE's security capabilities in that respect.

As a summary this shows that the security functions are not contradicting each other and are mutually supportive.

8.3.2 Assurance Measures Justification

The TOE summary specification in section 6.4 includes a justification that each TOE security assurance requirement is met by appropriate assurance measures.

8.3.3 Strength of function

The password mechanism used for authentication is one mechanism in the TSF that is implemented by a permutational or probabilistic mechanism subject to a strength of function analysis within the evaluation of this TOE. For the password based authentication mechanism of the security function IA.1, a minimum strength of SOF-medium is claimed. This is done in accordance with the SOF claim for the related security functional requirement FIA_SOS.1. This claim is consistent with the security objective O.AUTHORIZATION and the statement in section 3.2 which says that the TOE should „protect against threats of inadvertent or casual attempts to breach the system security". A highly skilled and well funded attacker is explicitly excluded from the threat scenario described in section 3.2.

The SOF-medium claim does **not** apply to the cryptographic algorithms, including the cryptographic properties of the cryptographic hash functions implemented in the TOE. Excluding cryptographic algorithms and related functions from the strength of function analysis is in compliance with the CEM, remarks on ASE_REQ.1.15, para 422.

Therefore, a strength of SOF-medium is consistent with the description of the TOE environment.

8.3.4 PP Threats

[LSPP] and [CAPP] do not state any threats, but derive their objectives from the Organisational Security Policies only. Therefore, no mapping is required for them.

For [RBACPP], the following mapping has been made:

Table 8-10: Mapping of ST threats to PP threats

ST	PP	Rationale
T.UAUSER An attacker (possibly, but not necessarily, an unauthorized user of the TOE) may impersonate an authorized user of the TOE. This includes the threat of an authorized user that tries to impersonate as another authorized user without knowing the authentication information.	**T.ENTRY** An unauthorized person may gain logical access to the TOE.	Since logical access to the TOE is only possible for authorized users, impersonation of an authorized user is the only means to gain logical access to the TOE. The two threats are therefore equivalent.
T.UAACCESS An authorized user of the TOE may access information resources without having permission from the person who owns, or is responsible for, the information resource for the type of access.	**T.ACCESS** A user may gain access to resources or perform operations for which no access rights have been granted	Both threats are concerned with access to objects without proper access rights having been granted by the use authorized to grant these rights. For the purpose of this ST, these threatse are identical.

All other threats from [RBACPP] have been added to the ST (restricting them to the LSPP/RBAC mode of operation, as [RBACPP] is only relevant in this mode).

8.3.5 PP Assumptions

The [RBACPP] assumptions are met by this ST as follows:

Table 8-11: Mapping of [RBACPP] assumptions

PP	ST	Rationale
A.ASSET	Statement on threat agents in section 3.2 and A.LOCATE	"It is also assumed that the value of the stored assets merits moderately intensive penetration or masquerading attacks." This first part of A.ASSET is covered by the statement on the threat agents in section 3.2. Note that [RBACPP] lists this statement as an assumption on the physical aspects. Therefore, the mentioned penetration or masquerading attacks apply to attacks trying to gain physical access to the TOE only.
A.LOCATE	A.LOCATE	Identical assumptions, despite slightly different wording
A.PROTECT	A.PROTECT	Identical assumptions, except for the restriction of [RBACPP] of this assumption for "potentially hostile outsiders".
A.ACCESS	A.ACCESS	Assumption copied from [RBACPP]
A.MANAGE	A.MANAGE	All aspects of the [RBACPP] assumption have been added to the ST assumption, which originates from CAPP/LSPP.
A.OWNER	A.OWNER	Assumption copied from [RBACPP]
A.CONNECT	A.CONNECT	Assumption in ST also requires all connections to be protected.

The assumptions for [LSPP] and [CAPP] have been copied to the ST and changed as follows:

- A.MANAGE has been augmented with the additional requirements on role management from [RBACPP]
- A.CONNECT has been changed to reflect the fact that the ST is capable to protect connections with encrypted tunnels, thus not requiring all connections to be within a physically controlled environment.
- additional assumptions from LSPP not found in CAPP apply to the LSPP/RBAC mode of operation only.

8.3.6 PP Objectives

The objectives in the ST have been copied from the various protection profiles and map to the objectives in the protection profiles as follows:

Table 8-12: Mapping of TOE objectives to the protection profiles

ST	CAPP	LSPP	RBAC
O.AUTHORIZATION	O.AUTHORIZATION	O.AUTHORIZATION	O.ACCOUNT, O.KNOWN, O.ENTRY
O.DISCRETIONARY_ACCESS	O.DISCRETIONARY_ACCESS	O.DISCRETIONARY_ACCESS	
O.MANDATORY_ACCESS		O.MANDATORY_ACCESS	
O.AUDITING	O.AUDITING	O.AUDITING	O.ACCOUNT, O.AUDIT
O.RESIDUAL_INFO	O.RESIDUAL_INFORMATION	O.RESIDUAL_INFORMATION	
O.MANAGE	O.MANAGE	O.MANAGE	O.ADMIN
O.ENFORCEMENT	O.ENFORCEMENT	O.ENFORCEMENT	
O.COMPROT			
O.DUTY			O.DUTY
O.HIERARCHICAL			O.HIERARCHICAL
O.ROLE			O.ROLE

While the LSPP and CAPP objectives are trivially met by the ST objectives, as the ST objectives have been copied from these PPs, some of the objectives in [RBACPP] have been replaced by other objectives, as they were either identical in contents or the objective was met by a combination of other objectives:

- O.ACCOUNT: accountability is provided by identification and authentication, which is embedded in O.AUTHORIZATION, and auditing, which is directly met by O.AUDITING
- O.ADMIN is directly covered by O.MANAGE
- O. AUDIT is somewhat more specific than O.AUDITING, but fully covered by it.
- O.ENTRY is a negated formulation of O.AUTHORIZATION
- O.KNOWN requires all users to be identified and authenticated, which is fully covered by O.AUTHORIZATION

All other RBAC objectives have been copied into the ST.

As Table 8-12 shows, there is only one additional security objective for the TOE (O.COMPROT), which has been defined to reflect the ability of the TOE to connect with trusted IT products via trusted channels.

With respect to the objectives for the TOE's environment, the mapping is as follows:

Table 8-13: Mapping of TOE's environment objectives to the protection profiles

ST	CAPP	LSPP	RBAC
OE.ADMIN			
OE.CREDEN	O.CREDEN	O.CREDEN	
OE.INSTALL	O.INSTALL	O.INSTALL	O.INSTALL
OE.PHYSICAL	O.PHYSICAL	O.PHYSICAL	O.PHYSICAL
OE.INFO_PROTECT			O.INSTALL, O.CONNECT
OE.MAINTENANCE			
OE.RECOVER			
OE.SOFTWARE_IN			O.INSTALL
OE.SERIAL_LOGIN			
OE.PROTECT			O.CONNECT

While the LSPP and CAPP objectives are trivially met by the ST objectives, as the ST objectives have been copied from these PPs, some of the objectives in [RBACPP] have been replaced by other objectives, as they were either identical in contents or the objective was met by a combination of other objectives:

- O.INSTALL: The objective of proper delivery and installation is met by OE.INSTALL and OE.SOFTWARE_IN. The additional requirement for proper operation of the TOE is met by the combination of OE.INFO_PROTECT, OE.RECOVER and OE.SOFTWARE_IN.
- O.CONNECT: The absence of connections that could undermine the system's security is provided by OE.PROTECT and the part of OE.INFO_PROTECT covering network and peripheral cabling. As the TOE allows connections over encrypted tunnels, the TOE objective O.COMPROT also contributes to this objective.

All other RBAC objectives are directly met by the corresponding ST objective as shown in Table 8-13.

As Table 8-13 shows, OE.ADMIN, OE.MAINTENANCE, OE.RECOVER and OE.SERIAL_LOGIN have been defined in addition to the environment objectives identified in the protection profiles. They have been added to this ST to allow for a more distinguished description of the TOE environment - this does not impact the conformance of this ST to the PPs.

8.3.7 PP SFRs

The following table shows which SFRs have been taken from which protection profile. SFRs with empty entries in all PP columns have been defined in addition to the protection profiles. In cases where the PP SFR has been replaced with a hierarchical superior one, the original SFR of the PP is mentioned in the column of the respective PP.

Table 8-14: TOE SFR mapping to protection profiles
Rows with SFRs only relevant in LSPP/RBAC mode have been colored.

SFR	Title	CAPP	LSPP	RBAC
FAU_GEN.1	Audit Data Generation	X	X	X
FAU_GEN.2	User Identity Association	X	X	X
FAU_SAR.1	Audit Review	X	X	X
FAU_SAR.2	Restricted Audit Review	X	X	X
FAU_SAR.3	Selectable Audit Review	X	X	X
FAU_SEL.1	Selective Audit	X	X	X
FAU_STG.1	Guarantees of Audit Data Availability	X	X	X
FAU_STG.3	Action in Case of Possible Audit Data Loss	X	X	
FAU_STG.4	Prevention of Audit Data Loss	X	X	
FCS_CKM.1(1)	Cryptographic key generation (SSL: Symmetric algorithms)			
FCS_CKM.1(2)	Cryptographic key generation (SSH: Symmetric algorithms)			
FCS_CKM.1(3)	Cryptographic key generation (SSL: RSA)			
FCS_CKM.2(1)	Cryptographic key distribution (SSL: RSA public keys)			
FCS_CKM.2(2)	Cryptographic key distribution (SSH: Diffie-Hellman key negotiation)			
FCS_CKM.2(3)	Cryptographic key distribution (SSH: DSS public keys)			
FCS_CKM.2(4)	Cryptographic key distribution (SSL: Symmetric keys)			
FCS_COP.1(1)	Cryptographic operation (RSA)			

SFR	Title	CAPP	LSPP	RBAC
FCS_COP.1(2)	Cryptographic operation (SSL: Symmetric operations)			
FCS_COP.1(3)	Cryptographic operation (SSH: Symmetric operations)			
FDP_ACC.1(1)	Discretionary Access Control Policy	X	X	
FDP_ACC.1(2) LSPP/RBAC mode	Role-based Access Control Policy			X
FDP_ACF.1(1)	Discretionary Access Control Functions	X	X	
FDP_ACF.1(2) LSPP/RBAC mode	Role-based Access Control Functions			X
FDP_ETC.1 LSPP/RBAC mode	Export of unlabeled user data		X	
FDP_ETC.2 LSPP/RBAC mode	Export of labeled user data		X	
FDP_IFC.1 LSPP/RBAC mode	Mandatory access control policy		X	
FDP_IFF.2 LSPP/RBAC mode	Mandatory access control functions		X	
FDP_ITC.1 LSPP/RBAC mode	Import of unlabeled user data		X	
FDP_ITC.2 LSPP/RBAC mode	Import of labeled user data		X	
FDP_RIP.2	Object Residual Information Protection	X	X	
Note 1	Subject Residual Information Protection	X	X	
FDP_UCT.1	Basic data exchange confidentiality			
FDP_UIT.1	Data exchange integrity			
FIA_ATD.1	User Attribute Definition	X	X	X
FIA_SOS.1	Strength of Authentication Data	X	X	
FIA_UAU.2	Authentication	FIA_UAU.1	FIA_UAU.1	X
FIA_UAU.7	Protected Authentication Feedback	X	X	
FIA_UID.2	Identification	FIA_UID.1	FIA_UID.1	X
FIA_USB.1	User-Subject Binding	X	X	X
FMT_MSA.1(1)	Management of Object Security Attributes	X	X	
FMT_MSA.1(2) LSPP/RBAC mode	Management of object security attributes for MAC		X	
FMT_MSA.1(3) LSPP/RBAC mode	Management of User Security Attributes			X
FMT_MSA.2	Secure security attributes			X
FMT_MSA.3(1)	Static Attribute Initialization for DAC	X	X	
FMT_MSA.3(2) LSPP/RBAC mode	Static Attribute Initialization for MAC		X	
FMT_MSA.3(3) LSPP/RBAC mode	Static Attribute Initialization for RBAC			X
FMT_MTD.1(1)	Management of the Audit Trail	X	X	
FMT_MTD.1(2)	Management of Audited Events	X	X	
FMT_MTD.1(3)	Management of User Attributes	X	X	

SFR	Title	CAPP	LSPP	RBAC
FMT_MTD.1(4)	Management of Authentication Data	X	X	
FMT_MTD.1(5) LSPP/RBAC mode	Management of RBAC TSF Data			X
FMT_MTD.3 LSPP/RBAC mode	Secure TSF Data			X
FMT_REV.1(1)	Revocation of User Attributes	X	X	X
FMT_REV.1(2)	Revocation of Object Attributes	X	X	
FMT_SMF.1	Specification of Management Functions			
FMT_SMR.2	Security Management Roles	FMT_SMR.1	FMT_SMR.1	X
FPT_AMT.1	Abstract Machine Testing	X	X	X
FPT_FLS.1 LSPP/RBAC mode	Failure with preservation of Secure State			X
FPT_RCV.1 LSPP/RBAC mode	Manual Recovery			X
FPT_RCV.4 LSPP/RBAC mode	Function Recovery			X
FPT_RVM.1	Reference Mediation	X	X	X
FPT_SEP.1	Domain Separation	X	X	X
FPT_STM.1	Reliable Time Stamps	X	X	X
FPT_TDC.1 LSPP/RBAC mode	Inter-TSF basic TSF data consistency			
FPT_TST.1 LSPP/RBAC mode	TSF Self Test			X
FTA_LSA.1 LSPP/RBAC mode	Limitation on Scope of Selectable Attributes			X
FTA_TSE.1 LSPP/RBAC mode	TOE Session Establishment			X
FTP_ITC.1	Inter-TSF trusted channel			

- In FAU_GEN.1, the RBAC requirements have been incorporated into Table 5-1 as follows:
  - "(a) Start-up and Shutdown of the audit functions" is fully covered by the events assigned to FAU_GEN.1
  - "(b) All auditable events for the basic level of audit" is covered by adding all events for the basic level of auditing to the table
  - "(c) (i) Assignment of Users, Roles and Privileges to Roles; (ii) Deletion of Users, Roles and Privileges from Roles (iii) Creation and Deletion of Roles" is covered by the events assigned to FMT_MTD.1(5).
- LSPP: FMT_MSA.1, FMT_MSA.3 contain impermissible "inline" iterations. They have been separated into correct iterations FMT_MSA.1(1), FMT_MSA.1(2), FMT_MSA.3(1) and FMT_MSA.3(2.)
- FIA_USB.1: Note2 from LSPP and CAPP has been incorporated into the SFR by newer versions of the CC.
- FMT_SMF.1 has been added to comply with the current version of [CC], which defines dependencies of two security functional requirements (FMT_MSA.1 and FMT_MTD.1) included in the PPs.
- FPT_TDC.1 has been added to fulfil a dependency to
- FTP_ITC.1 has been added to fulfil a dependency to FDP_ITC.2 (missing in [LSPP]), as well as FDP_UCT.1 and FDB_UIT.1

All security functional requirements in this ST are inherited from one of the protection profiles and the operations allowed / required by one or more of the PPs are performed and indicated in bold letters.

As table Table 8-14 shows, FCS_CKM.1(1), FCS_CKM.1(2), FCS_CKM.1(3), FCS_CKM.2(1), FCS_CKM.2(2), FCS_CKM.2(3), FCS_CKM.2(4), FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3), FDP_UCT.1, FDP_UIT.1, FMT_SMF.1, FPT_TDC.1 and FTP_ITC.1 have been added to this ST in addition to the SFRs required by the protection profiles.

Additional SFRs for the TOE IT environment have been defined to cope with the more distinguished description of the TOE environment - this does not impact the conformance of this ST to the PP.

9 Abbreviations

ACL	Access Control List
AIX	Advanced Interactive Executive
ANSI	American National Standards Institute
CAPP	Controlled Access Protection Profile
CC	Common Criteria
CD	Compact Disc
CPU	Central Processing Unit
DAC	Discretionary Access Control
DVD	Digital Versatile Disc
FPR	Floating Point Register
FSO	File System Object
FTP	File Transfer Protocol
GPR	General Purpose Register
ID	Identifier
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
IPC	Inter-Process Communication
LAN	Local Area Network
ISO	International Standards Organization
MD5	Message Digest 5
PAM	Pluggable Authentication Module
PDF	Portable Data Format
PP	Protection Profile
RHEL	Red Hat Enterprise Linux
SSH	Secure Shell
ST	Security Target
TCP	Transmission Control Protocol
TOE	Target of Evaluation
TSF	TOE Security Functions
UDP	User Datagram Protocol
VFS	Virtual File System
VMM	Virtual Memory Manager