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

Version: 1.12

Last Update: 2007-06-06

Table of Content

Document History

Version	Date	Changes	Summary	Author
1.0	2005-07-25	no	Initial Version	Helmut Kurth, atsec
1.1	2005-08-15	various	Addressing comments from IBM and inconsistencies identified by the author. Including IPv6 and IPsec.	Helmut Kurth, atsec
1.2	2005-11-21	various	Addressing comments from IBM and the evaluator.	Klaus Weidner, atsec
1.3	2005-11-22	minor	Removed iSeries from hardware list, minor fixes	Klaus Weidner, atsec
1.4	2006-07-14	various	Added SSH and SSL standards compliance tables. Updated role privileges, audit table. Removed IPSec crypto mappings.	Klaus Weidner, atsec
1.5	2006-09-08	various	Clarifications and updates based on evaluator feedback	Klaus Weidner, atsec
1.6	2006-10-11	minor	Updated trusted databases list, minor other updates	Klaus Weidner, atsec
1.7	2007-01-21	various	Changes to reflect ongoing development; and updates based on evaluator feedback	Klaus Weidner, atsec
1.8	2007-03-27	various	Updates based on evaluator feedback; corrections and clarifications	Klaus Weidner, atsec
1.9	2007-04-24	various	Update trusted databases list, RPM package information, corrections and clarifications	Klaus Weidner, atsec
1.10	2007-05-01	minor	add aide config file and database to trusted databases list; final RPM package list	Klaus Weidner, atsec
1.11	2007-05-23	minor	minor clarifications based on evaluator feedback	Klaus Weidner, atsec
1.12	2007-06-06	minor	CC version used is 2.3	Klaus Weidner, atsec

References

[CC] Common Criteria for Information Technology Security Evaluation, Version 2.3, August 2005, Part 1 to 3

[CEM] Common Methodology for Information Technology Security Evaluation, Version 2.3, August 2005

[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

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

[LSPP] Labeled Security Protection Profile, Issue 1.b, 8 October 1999

[RBACPP] Role-Based Access Control Protection Profile, Version 1.0, July 30, 1998

[SSLv3] The SSL Protocol Version 3.0, http://wp.netscape.com/eng/ssl3/draft302.txt

[SSH-AUTH] RFC 4252: The Secure Shell (SSH) Authentication Protocol, http://www.ietf.org/rfc/rfc4252.txt

[SSH-TRANS] RFC 4253: The Secure Shell (SSH) Transport Layer Protocol, http://www.ietf.org/rfc/rfc4253.txt

[HMAC] RFC 2104: HMAC: Keyed-Hashing for Message Authentication, http://www.ietf.org/rfc/rfc2104.txt

[TLS-AES] RFC 3268: Advanced Encryption Standard (AES) Ciphersuites for Transport Layer Security (TLS), http://www.ietf.org/rfc/rfc3268.txt

[IKE] RFC 2409: The Internet Key Exchange (IKE), http://www.ietf.org/rfc/rfc2409.txt

[IPSEC] RFC 2401: Security Architecture for the Internet Protocol, http://www.ietf.org/rfc/rfc2101.txt

[TARGET] Red Hat Enterprise Linux 5 Security Target for CAPP, RBAC and LSPP compliance (this document)

[X.509] ITU-T RECOMMENDATION X.509 | ISO/IEC 9594-8: INFORMATION TECHNOLOGY - OPEN SYSTEMS INTERCONNECTION - THE DIRECTORY: PUBLIC-KEY AND ATTRIBUTE CERTIFICATE FRAMEWORKS

1 Introduction

This is version 1.12 of the Security Target document for the evaluation of Red Hat Enterprise Linux 5 Server and Red Hat Enterprise Linux 5 Client. Both products are identical with respect to the code base and the user documentation. The only differences between the two products lines are:

- The Server product is available on a wide range of hardware platforms while the Client product is available on the Intel x86 and Intel Itanium and AMD64 platforms only. This evaluation covers the Server and Client products on all supported IBM platforms as applicable.
- The Client product includes additional packages that are not subject to this evaluation and have been excluded from the evaluated configuration. In the evaluated configuration, both the Client and Server products are configured for "server" use, without a graphical desktop.
- The Server product comes with a significantly higher level of support. This aspect is not relevant for the evaluation.

Except for the package that displays the release (redhat-release-*), the Client and Server product should be configured with the identical set of packages within this evaluation and therefore do not significantly differ in the overall functionality as well as the security functions provided.

This Security Target has been derived from the Security Target used for the previous evaluation of Red Hat Enterprise Linux at the EAL4+ level and CAPP compliance, also sponsored by IBM.

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

1.1 ST Identification

Title: Red Hat Enterprise Linux Version 5 Security Target for CAPP, RBAC and LSPP Compliance, Version 1.12

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

This document is the security target for the CC evaluation of the Red Hat Enterprise Linux 5 Server and Red Hat Enterprise Linux 5 Client operating system products, and is conformant to the Common Criteria for Information Technology Security Evaluation [CC] with extensions as defined in the Controlled Access Protection Profile [CAPP], the Role-based Access Control Protection Profile [RBACPP] 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 Server and Red Hat Enterprise Linux 5 Client operating system (In the rest of this document we will use the term "Red Hat Enterprise Linux" as a synonym for this).

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. It also meets all of the requirements of the Controlled Access Protection Profile, the Role-based Access Control Protection Profile and the Labeled Security Protection Profile developed by the Information Systems Security Organization within the National Security Agency to map the TCSEC C2 and B1 classes of the U.S. Department of Defence (DoD) Trusted Computer System Evaluation Criteria (TCSEC) to the Common Criteria framework. This Security Target therefore claims full compliance with the requirements of those Protection Profiles and also includes additional functional and assurance packages beyond those required by [CAPP], [RBACPP] and [LSPP].

The TOE can operate in two different modes of operation called "CAPP mode" and "LSPP mode". In CAPP mode the SELinux security module does not enforce a mandatory access control policy and does not recognize sensitivity labels of subjects and objects. SELinux can either be disabled completely, or enabled with a non-MLS policy such as the "targeted" or "strict" policies which only add additional restrictions to the CAPP requirements without interfering with the "root" administrator role. In this mode the TOE enforces all security requirements of [CAPP] but does not enforce the requirements of [LSPP] and [RBACPP].

In LSPP mode the SELinux security module is configured to enforce the mandatory access control policy based on the labels of subjects and objects as required by [LSPP], and the requirements of [RBACPP]. Note that a system in LSPP mode can optionally be configured to use a single sensitivity label for all subjects and objects to provide an operational mode equivalent to pure RBAC with no mandatory access control.

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.

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, using CC version 2.3.

The extensions to part 2 of the Common Criteria are those introduced by the Controlled Access Protection Profile [CAPP] (which are also included in the Labeled Securtity Protection Profile [LSPP]).

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.

Administrative User: This term refers to a user in one of the defined administrative roles of a Red Hat Enterprise Linux system. The TOE defines a set of administrative roles where each role has specific administrative authoritities. Splitting the administrative authorities among different roles allows for a more controlled operational environment without the need for a single user to have all administrative authorities.

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.

Classification: A sensitivity label associated with an object.

Clearance: A sensitivity label associated with a subject or user.

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

Dominate: Sensitivity label A *dominates* sensitivity label B if the hierarchical level of A is greater than or equal to the hierarchical level of B, and the category set of label A is a proper subset of or equal to the category set of label B. (cf. *Incomparable* sensitivity labels)

Incomparable: Security labels A and B are *incomparable* if A does not dominate B and B does not dominate A, for example if neither of their category sets is a subset of the other.

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

Named Object: In Red Hat Enterprise Linux, those objects that are subject to discretionary, role based or mandatory access control. This includes all objects except memory objects.

Named Object Security Attributes: In Red Hat Enterprise Linux those attributes are the object type and (in LSPP mode) the sensitivity label of the object.

Object: In Red Hat Enterprise Linux, objects belong to one of the following categories: file system objects, IPC objects, memory objects, and network objects. Processes are objects when they are the target of signal-related system calls.

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

Role: A role represents a set of actions that an authorized user, upon assuming the role, can perform.

Sensitivity Label: When operated in LSPP mode the TOE attaches a sensitivity label to each named object. This label consists of a hierarchical sensitivity level and a set of zero or more categeories. In LSPP mode the policy defines the number and names of the sensitivity levels and categories.

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 for a TOE that claims FPT_SEP.1 and/or FPT_RVM.1 and relies on hardware / firmware functions to implement this.

Type: The TOE allows to assign a defined type to a subject (process) and to an object and enforce access control based on those types. Types are used to model role based access control.

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

User 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; and user roles. In LSPP mode this also includes the user clearance which defines the maximum sensitivity label a user can have access to.

2 TOE Description

The target of evaluation (TOE) is the operating system Red Hat Enterprise Linux 5 Server and Red Hat Enterprise Linux 5 Client product (also referred to in this document as “Red Hat Enterprise Linux”).

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 and and small server type computer systems.

The Red Hat Enterprise Linux evaluation covers a potentially distributed, but closed network of IBM System x, System p5, System z, and eServer servers running the evaluated versions and configurations 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, the BootProm firmware and potentially other firmware layers between the hardware and Red Hat Enterprise Linux are considered to be part of the TOE.

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

The TOE includes standard networking applications, such as ftp, stunnel and ssh. It also supports the use of IPsec for exchange of labeled data.

System administration tools include the standard 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 network server using a port above 1024 may be used as a normal application running without root privileges on top of the TOE. The additional documentation specific for the evaluated configuration provides guidance how to set up such applications on the TOE in a secure way.

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 can be configured to operate in one of two modes, *CAPP mode* and *LSPP mode,* as defined in section 1.2 of this document.

In LSPP mode, the TOE uses mandatory access control together with discretionary and role-based access control. In LSPP mode rules are defined to assign *sensitivity labels* to subjects and objects and to implement the information flow mandatory access control policy defined in [LSPP].

In LSPP mode, administrative actions are delegated to specific roles. Any user in a role that is allowed to perform administrative actions is considered an administrative user. In addition the TOE supports types that can be associated with objects and domains that can be associated with processes. Roles are defined by the domains they have access to. A predefined policy file, which is part of the TOE configuration, defines the rules between domains and types. With this definition of roles and the access rights implied by the individual roles the TOE complies with the requirements of [RBACPP].

The TOE is intended to operate in a networked environment with other instantiations of the TOE as well as other well-behaved peer 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. 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 together with the attributes and roles assigned to the user as the basis for access control decisions. The TOE authenticates the claimed identity of the user before allowing the user to perform any further actions. The TOE internally maintains a set of identifiers associated with processes, which are derived from the unique user identifier upon login of the user. Some of those identifiers may change during the execution of the process according to a policy implemented by the TOE.

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, by administrative users, by the object type and the object sensitivity label. Ownership of named objects may be transferred under the control of the access control policy.

Discretionary 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. In LSPP mode mandatory access control can be used to prohibit direct flow of information to objects that have a sensitivity label dominating the security label of the subject as well as to objects that have a security label that is incompatible with the security label of the subject.

In LSPP mode, the TOE enforces a mandatory access control policy based on sensitivity labels that are attached to objects managed by the TOE. The mechanisms to attach those labels to the objects and assign intial values to those labels are implemented in the SELinux security module which extends the security mechanisms of the Linux kernel using the loadable securtity module feature. SELinux provides a flexible way to define security policies to be enforced for subjects and objects within the kernel. This evaluation includes a specific policy defined to address the requirements of the Labeled Security Protection Profile [LSPP] and the roles and privileges required to manage this policy efficiently.

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

- Access Control Lists,
- Domains and type enforcement
- Labels assigned to a number of kernel objects within a security context defined and managed by the SELinux security module (LSPP mode only),
- A Journaling File System (ext3),

  Integrated authentication framework (PAM).
- 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 hardware check functions. 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 TOE are:

- Identification and Authentication
- Audit
- Discretionary Access Control
- Mandatory Access Control (LSPP mode only)
- Role-Based Access Control (LSPP 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 root login to specific terminals are also included. When operating in LSPP mode users may select the sensitivity label of their session from a range of labels allowed for them to use.

The TSF software enforces restrictions when establishing user sessions to ensure that the set of active roles available to that user is limited to those roles for which the user is authorized, and ensures that sessions can only be established with a nonempty set of active roles.

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 system 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.

The audit system also records the sensitivity labels of subjects and objects as well as the role that has allowed access when the TOE operates in LSPP mode.

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.

IPC objects use permission bits for discretionary access control.

2.2.4 Mandatory Access Control (LSPP mode only)

Mandatory access control (MAC) restricts access to file system objects, IPC objects and network objects based on labels attached to those objects as part of their security context managed by SELinux. The label is compared to the security label of the subject that attempts to access/use the object. The mandatory access control includes a fixed set of rules based on the labels of the subject and the object and the type of access attempted that determine if the subject may access the object in the attempted way. Mandatory access control checks are performed in addition to the discretionary access control checks and access is granted only if access is granted by both the mandatory and the discretionary access control policies.

2.2.5 Role-Based Access Control (LSPP mode only)

Roles in the TOE are defined via types and access to types. A “type” is a security attribute given to an object or a process. The type of a process is commonly called a “domain”. Policy rules define how domains may interact with objects and with other domains.

Roles can be assigned to users and define which user can have access to which domain. A user may have several roles assigned to him but will always act in one role only. To change from his current role to another role that has been assigned to him he needs to use the newrole command which requires re-authentication. This prohibits that the user’s role is changed by a malicious program without the user knowing this. In addition the transition between roles may be restricted by the policy.

The TOE has a hierarchical set of roles defined in the policy. Those are:

- Root administrator: This is the classical superuser role which is hierarchical to all other roles

- System process: This is a role that should be assigned to specific system processes like daemons
- System administrator: This is a role for general system administration
- Security administrator: This is a role for the administration of security (policy and security contexts)
- Staff: This is a user role for users allowed to use the *newrole* and *su* commands
- User: This is a general user role without being allowed to use the *newrole* and *su* commands
- Audit administrator: This is a role for administration of the audit policy and the evaluation of audit records

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 root privileges are used for system management. 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.

In CAPP and LSPP modes security management can be split between different roles.

2.2.8 Secure Communication

The TOE supports secure communication with other systems via the SSH v2.0 and SSL v3 protocol. 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 protocol 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 and (in LSPP mode) 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 (in LSPP mode) MAC permissions.

The TOE including the hardware and firmware components are required to be physically protected from unauthorized access. The system kernel mediates all access to hardware components that are protected from direct access by user programs. A user process may execute unprivileged instructions and read or write to memory and processor regsister within the bounds defined by the kernel for the user process without those types of access being 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 Server
- Red Hat Enterprise Linux 5 Client

In the evaluated configuration, both the Client and Server products are configured for “server” use, without a graphical desktop.

The TOE and its documentation are supplied on CD-ROM except for the update which needs to be downloaded from the Red Hat web site. Updates are delivered via RedHat's up2date client. This package contains the additional user and administrator documentation, all packages that have been updated to fix problems 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 shows the packages that make 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 setuid or setgid to root.

The list contains the packages with their version numbers. In a few cases the version numbers of packages may be different for different platforms.

	rpm-ppc64.lst	rpm-s390x.lst	rpm-x86_64.lst	Source
Deployment_Guide-en-US	5.0.0-19.noarch	5.0.0-19.noarch	5.0.0-19.noarch	Deployment_Guide
GConf2	2.14.0-9.el5.ppc	-	2.14.0-9.el5.x86_64	GConf2
MAKEDEV	3.23-1.2.ppc	3.23-1.2.s390x	3.23-1.2.x86_64	MAKEDEV
NetworkManager	0.6.4-6.el5.ppc	-	0.6.4-6.el5.x86_64	NetworkManager
ORBit2	2.14.3-4.el5.ppc	-	2.14.3-4.el5.x86_64	ORBit2
OpenIPMI	2.0.6-5.el5.3.ppc	2.0.6-5.el5.3.s390x	2.0.6-5.el5.3.x86_64	OpenIPMI
OpenIPMI-libs	2.0.6-5.el5.3.ppc	2.0.6-5.el5.3.s390x	2.0.6-5.el5.3.x86_64	OpenIPMI
SysVinit	2.86-14.ppc	2.86-14.s390x	2.86-14.x86_64	SysVinit
acl	2.2.39-2.1.el5.ppc	2.2.39-2.1.el5.s390x	2.2.39-2.1.el5.x86_64	acl
acpid	-	-	1.0.4-5.x86_64	acpid
aide	0.12-9.el5.ppc	0.12-9.el5.s390x	0.12-9.el5.x86_64	aide
amtu	1.0.4-4.ppc	1.0.4-4.s390x	1.0.4-4.x86_64	amtu
anacron	2.3-45.el5.ppc	2.3-45.el5.s390x	2.3-45.el5.x86_64	anacron
aspell	0.60.3-7.1.ppc	0.60.3-7.1.s390	0.60.3-7.1.i386	aspell
aspell#2	0.60.3-7.1.ppc64	0.60.3-7.1.s390x	0.60.3-7.1.x86_64	aspell
aspell-en	6.0-2.1.ppc	6.0-2.1.s390x	6.0-2.1.x86_64	aspell-en
at	3.1.8-82.fc6.ppc	3.1.8-82.fc6.s390x	3.1.8-82.fc6.x86_64	at
atk	1.12.2-1.fc6.ppc	-	1.12.2-1.fc6.x86_64	atk
attr	2.4.32-1.1.ppc	2.4.32-1.1.s390x	2.4.32-1.1.x86_64	attr
audit	1.3.1-6.el5.ppc	1.3.1-6.el5.s390x	1.3.1-6.el5.x86_64	audit
audit-libs	1.3.1-6.el5.ppc	1.3.1-6.el5.s390	1.3.1-6.el5.i386	audit
audit-libs#2	1.3.1-6.el5.ppc64	1.3.1-6.el5.s390x	1.3.1-6.el5.x86_64	audit
audit-libs-devel	1.3.1-6.el5.ppc	1.3.1-6.el5.s390	1.3.1-6.el5.i386	audit
audit-libs-devel#2	1.3.1-6.el5.ppc64	1.3.1-6.el5.s390x	1.3.1-6.el5.x86_64	audit
audit-libs-python	1.3.1-6.el5.ppc	1.3.1-6.el5.s390x	1.3.1-6.el5.x86_64	audit
authconfig	5.3.12-2.el5.ppc	5.3.12-2.el5.s390x	5.3.12-2.el5.x86_64	authconfig
autoconf	2.59-12.noarch	2.59-12.noarch	2.59-12.noarch	autoconf
autofs	5.0.1-0.rc2.42.ppc	5.0.1-0.rc2.42.s390x	5.0.1-0.rc2.42.x86_64	autofs
automake	1.9.6-2.1.noarch	1.9.6-2.1.noarch	1.9.6-2.1.noarch	automake
basesystem	8.0-5.1.1.noarch	8.0-5.1.1.noarch	8.0-5.1.1.noarch	basesystem
bash	3.1-16.1.ppc	3.1-16.1.s390x	3.1-16.1.x86_64	bash
bc	1.06-21.ppc	1.06-21.s390x	1.06-21.x86_64	bc
beecrypt	4.1.2-10.1.1.ppc	4.1.2-10.1.1.s390x	4.1.2-10.1.1.x86_64	beecrypt
bind-libs	9.3.3-7.el5.ppc	9.3.3-7.el5.s390x	9.3.3-7.el5.x86_64	bind
bind-libs#2	9.3.3-7.el5.ppc64	-	-	bind
bind-utils	9.3.3-7.el5.ppc	9.3.3-7.el5.s390x	9.3.3-7.el5.x86_64	bind
binutils	2.17.50.0.6-2.el5.ppc	2.17.50.0.6-2.el5.s390x	2.17.50.0.6-2.el5.x86_64	binutils
bison	2.3-2.1.ppc	2.3-2.1.s390x	2.3-2.1.x86_64	bison
bluez-gnome	0.5-5.fc6.ppc	-	0.5-5.fc6.x86_64	bluez-gnome
bluez-libs	3.7-1.ppc	-	3.7-1.x86_64	bluez-libs
bluez-libs#2	3.7-1.ppc64	-	-	bluez-libs
bluez-utils	3.7-2.ppc	-	3.7-2.x86_64	bluez-utils
bzip2	1.0.3-3.ppc	1.0.3-3.s390x	1.0.3-3.x86_64	bzip2
bzip2-libs	1.0.3-3.ppc	1.0.3-3.s390x	1.0.3-3.x86_64	bzip2
cairo	1.2.4-1.fc6.ppc	1.2.4-1.fc6.s390x	1.2.4-1.fc6.x86_64	cairo
ccid	1.0.1-6.el5.ppc	-	1.0.1-6.el5.x86_64	ccid
checkpolicy	1.33.1-2.el5.ppc	1.33.1-2.el5.s390x	1.33.1-2.el5.x86_64	checkpolicy
chkconfig	1.3.30.1-1.ppc	1.3.30.1-1.s390x	1.3.30.1-1.x86_64	chkconfig
chkfontpath	1.10.1-1.1.ppc	1.10.1-1.1.s390x	1.10.1-1.1.x86_64	chkfontpath
conman	0.1.9.2-4.el5.ppc	0.1.9.2-4.el5.s390x	0.1.9.2-4.el5.x86_64	conman
coolkey	1.0.1-16.el5.ppc	-	1.0.1-16.el5.i386	coolkey
coolkey#2	1.0.1-16.el5.ppc64	-	1.0.1-16.el5.x86_64	coolkey
coreutils	5.97-12.1.el5.ppc	5.97-12.1.el5.s390x	5.97-12.1.el5.x86_64	coreutils
cpio	2.6-20.ppc	2.6-20.s390x	2.6-20.x86_64	cpio
cpp	4.1.1-52.el5.ppc	4.1.1-52.el5.s390x	4.1.1-52.el5.x86_64	gcc
cpuspeed	1.2.1-1.45.el5.ppc	-	1.2.1-1.45.el5.x86_64	cpuspeed
cracklib	2.8.9-3.1.ppc	2.8.9-3.1.s390	2.8.9-3.1.i386	cracklib
cracklib#2	2.8.9-3.1.ppc64	2.8.9-3.1.s390x	2.8.9-3.1.x86_64	cracklib
cracklib-dicts	2.8.9-3.1.ppc	2.8.9-3.1.s390x	2.8.9-3.1.x86_64	cracklib
crash	4.0-3.14.ppc64	4.0-3.14.s390x	4.0-3.14.x86_64	crash
crontabs	1.10-8.noarch	1.10-8.noarch	1.10-8.noarch	crontabs
cryptsetup-luks	1.0.3-2.2.el5.ppc	1.0.3-2.2.el5.s390	1.0.3-2.2.el5.i386	cryptsetup-luks
cryptsetup-luks#2	1.0.3-2.2.el5.ppc64	1.0.3-2.2.el5.s390x	1.0.3-2.2.el5.x86_64	cryptsetup-luks
cups	1.2.4-11.8.el5.ppc	1.2.4-11.8.el5.s390x	1.2.4-11.8.el5.x86_64	cups
cups-libs	1.2.4-11.8.el5.ppc	1.2.4-11.8.el5.s390	1.2.4-11.8.el5.i386	cups
cups-libs#2	1.2.4-11.8.el5.ppc64	1.2.4-11.8.el5.s390x	1.2.4-11.8.el5.x86_64	cups
curl	7.15.5-2.el5.ppc	7.15.5-2.el5.s390x	7.15.5-2.el5.x86_64	curl
cvs	1.11.22-5.el5.ppc	1.11.22-5.el5.s390x	1.11.22-5.el5.x86_64	cvs
cyrus-sasl	2.1.22-4.ppc64	2.1.22-4.s390x	2.1.22-4.x86_64	cyrus-sasl
cyrus-sasl-devel	2.1.22-4.ppc	2.1.22-4.s390x	2.1.22-4.x86_64	cyrus-sasl
cyrus-sasl-lib	2.1.22-4.ppc	2.1.22-4.s390	2.1.22-4.i386	cyrus-sasl
cyrus-sasl-lib#2	2.1.22-4.ppc64	2.1.22-4.s390x	2.1.22-4.x86_64	cyrus-sasl
cyrus-sasl-plain	2.1.22-4.ppc	2.1.22-4.s390	2.1.22-4.i386	cyrus-sasl
cyrus-sasl-plain#2	2.1.22-4.ppc64	2.1.22-4.s390x	2.1.22-4.x86_64	cyrus-sasl
db4	4.3.29-9.fc6.ppc	4.3.29-9.fc6.s390	4.3.29-9.fc6.i386	db4
db4#2	4.3.29-9.fc6.ppc64	4.3.29-9.fc6.s390x	4.3.29-9.fc6.x86_64	db4
dbus	1.0.0-6.el5.ppc	1.0.0-6.el5.s390x	1.0.0-6.el5.x86_64	dbus
dbus#2	1.0.0-6.el5.ppc64	-	-	dbus
dbus-glib	0.70-5.ppc	0.70-5.s390x	0.70-5.x86_64	dbus-glib
dbus-glib#2	0.70-5.ppc64	-	-	dbus-glib
dbus-python	0.70-7.el5.ppc	0.70-7.el5.s390x	0.70-7.el5.x86_64	dbus-python
desktop-file-utils	0.10-7.ppc	0.10-7.s390x	0.10-7.x86_64	desktop-file-utils
device-mapper	1.02.13-1.el5.ppc	1.02.13-1.el5.s390	1.02.13-1.el5.i386	device-mapper
device-mapper#2	1.02.13-1.el5.ppc64	1.02.13-1.el5.s390x	1.02.13-1.el5.x86_64	device-mapper
dhcdbd	2.2-1.el5.ppc	-	2.2-1.el5.x86_64	dhcdbd
dhclient	3.0.5-3.el5.ppc	3.0.5-3.el5.s390x	3.0.5-3.el5.x86_64	dhcp
dhcpv6_client	0.10-33.el5.ppc	0.10-33.el5.s390x	0.10-33.el5.x86_64	dhcpv6
diffutils	2.8.1-15.2.2.ppc	2.8.1-15.2.2.s390x	2.8.1-15.2.2.x86_64	diffutils
dmidecode	-	-	2.7-1.28.2.el5.x86_64	dmidecode
dmraid	1.0.0.rc13-2.el5.ppc	1.0.0.rc13-2.el5.s390x	1.0.0.rc13-2.el5.x86_64	dmraid
dos2unix	3.1-27.1.ppc	3.1-27.1.s390x	3.1-27.1.x86_64	dos2unix
dosfstools	2.11-6.2.el5.ppc	2.11-6.2.el5.s390x	2.11-6.2.el5.x86_64	dosfstools
dump	0.4b41-2.fc6.ppc	0.4b41-2.fc6.s390x	0.4b41-2.fc6.x86_64	dump
e2fsprogs	1.39-8.el5.ppc	1.39-8.el5.s390x	1.39-8.el5.x86_64	e2fsprogs
e2fsprogs-devel	1.39-8.el5.ppc64	1.39-8.el5.s390x	1.39-8.el5.x86_64	e2fsprogs
e2fsprogs-libs	1.39-8.el5.ppc	1.39-8.el5.s390	1.39-8.el5.i386	e2fsprogs
e2fsprogs-libs#2	1.39-8.el5.ppc64	1.39-8.el5.s390x	1.39-8.el5.x86_64	e2fsprogs
ed	0.2-38.2.2.ppc	0.2-38.2.2.s390x	0.2-38.2.2.x86_64	ed
eject	2.1.5-4.2.el5.ppc	-	2.1.5-4.2.el5.x86_64	eject
elfutils	0.125-3.el5.ppc	0.125-3.el5.s390x	0.125-3.el5.x86_64	elfutils
elfutils-libelf	0.125-3.el5.ppc	0.125-3.el5.s390x	0.125-3.el5.x86_64	elfutils
elfutils-libs	0.125-3.el5.ppc	0.125-3.el5.s390x	0.125-3.el5.x86_64	elfutils
elinks	0.11.1-5.1.el5.ppc	0.11.1-5.1.el5.s390x	0.11.1-5.1.el5.x86_64	elinks
ethtool	5-1.el5.ppc	5-1.el5.s390x	5-1.el5.x86_64	ethtool
expat	1.95.8-8.2.1.ppc	1.95.8-8.2.1.s390x	1.95.8-8.2.1.i386	expat
expat#2	1.95.8-8.2.1.ppc64	-	1.95.8-8.2.1.x86_64	expat
expect	5.43.0-5.1.ppc	5.43.0-5.1.s390	5.43.0-5.1.i386	expect
expect#2	5.43.0-5.1.ppc64	5.43.0-5.1.s390x	5.43.0-5.1.x86_64	expect
expect-devel	5.43.0-5.1.ppc	5.43.0-5.1.s390	5.43.0-5.1.i386	expect
expect-devel#2	5.43.0-5.1.ppc64	5.43.0-5.1.s390x	5.43.0-5.1.x86_64	expect

fbset	2.1-22.ppc	-	2.1-22.x86_64	fbset
file	4.17-8.ppc	4.17-8.s390x	4.17-8.x86_64	file
filesystem	2.4.0-1.ppc	2.4.0-1.s390x	2.4.0-1.x86_64	filesystem
findutils	4.2.27-4.1.ppc	4.2.27-4.1.s390x	4.2.27-4.1.x86_64	findutils
finger	0.17-32.2.1.1.ppc	0.17-32.2.1.1.s390x	0.17-32.2.1.1.x86_64	finger
firstboot-tui	1.4.27.2-1.el5.noarch	-	1.4.27.2-1.el5.noarch	firstboot
flex	2.5.4a-41.fc6.ppc	2.5.4a-41.fc6.s390x	2.5.4a-41.fc6.x86_64	flex
fontconfig	2.4.1-6.el5.ppc	2.4.1-6.el5.s390x	2.4.1-6.el5.x86_64	fontconfig
fontconfig#2	2.4.1-6.el5.ppc64	-	-	fontconfig
freetype	2.2.1-16.el5.ppc	2.2.1-16.el5.s390x	2.2.1-16.el5.x86_64	freetype
freetype#2	2.2.1-16.el5.ppc64	-	-	freetype
ftp	0.17-33.fc6.ppc	0.17-33.fc6.s390x	0.17-33.fc6.x86_64	ftp
gawk	3.1.5-14.el5.ppc	3.1.5-14.el5.s390x	3.1.5-14.el5.x86_64	gawk
gcc	4.1.1-52.el5.ppc	4.1.1-52.el5.s390x	4.1.1-52.el5.x86_64	gcc
gcc-c++	4.1.1-52.el5.ppc	4.1.1-52.el5.s390x	4.1.1-52.el5.x86_64	gcc
gdbm	1.8.0-26.2.1.ppc	1.8.0-26.2.1.s390x	1.8.0-26.2.1.x86_64	gdbm
gdbm#2	1.8.0-26.2.1.ppc64	-	-	gdbm
gettext	0.14.6-4.el5.ppc64	0.14.6-4.el5.s390x	0.14.6-4.el5.x86_64	gettext
ghostscript	8.15.2-9.1.el5.ppc	8.15.2-9.1.el5.s390	8.15.2-9.1.el5.i386	ghostscript
ghostscript#2	8.15.2-9.1.el5.ppc64	8.15.2-9.1.el5.s390x	8.15.2-9.1.el5.x86_64	ghostscript
glib2	2.12.3-2.fc6.ppc	2.12.3-2.fc6.s390x	2.12.3-2.fc6.x86_64	glib2
glib2#2	2.12.3-2.fc6.ppc64	-	-	glib2
glib2-devel	2.12.3-2.fc6.ppc64	2.12.3-2.fc6.s390x	2.12.3-2.fc6.x86_64	glib2
glibc	2.5-12.ppc	2.5-12.s390	2.5-12.i686	glibc
glibc#2	2.5-12.ppc64	2.5-12.s390x	2.5-12.x86_64	glibc
glibc-common	2.5-12.ppc	2.5-12.s390x	2.5-12.x86_64	glibc
glibc-devel	2.5-12.ppc	2.5-12.s390	2.5-12.i386	glibc
glibc-devel#2	2.5-12.ppc64	2.5-12.s390x	2.5-12.x86_64	glibc
glibc-headers	2.5-12.ppc	2.5-12.s390x	2.5-12.x86_64	glibc
gnupg	1.4.5-12.ppc	1.4.5-12.s390x	1.4.5-12.x86_64	gnupg
gnutls	1.4.1-2.ppc	1.4.1-2.s390	1.4.1-2.i386	gnutls
gnutls#2	1.4.1-2.ppc64	1.4.1-2.s390x	1.4.1-2.x86_64	gnutls
gpm	1.20.1-74.1.ppc	1.20.1-74.1.s390	1.20.1-74.1.i386	gpm
gpm#2	1.20.1-74.1.ppc64	1.20.1-74.1.s390x	1.20.1-74.1.x86_64	gpm
grep	2.5.1-54.2.el5.ppc	2.5.1-54.2.el5.s390x	2.5.1-54.2.el5.x86_64	grep
groff	1.18.1.1-11.1.ppc	1.18.1.1-11.1.s390x	1.18.1.1-11.1.x86_64	groff
grub	-	-	0.97-13.x86_64	grub
gtk2	2.10.4-16.el5.ppc	-	2.10.4-16.el5.x86_64	gtk2
gzip	1.3.5-9.el5.ppc	1.3.5-9.el5.s390x	1.3.5-9.el5.x86_64	gzip
hal	0.5.8.1-19.el5.ppc	0.5.8.1-19.el5.s390x	0.5.8.1-19.el5.x86_64	hal
hal#2	0.5.8.1-19.el5.ppc64	-	-	hal
hdparm	6.6-2.ppc	-	6.6-2.x86_64	hdparm
hesiod	3.1.0-8.ppc	3.1.0-8.s390x	3.1.0-8.x86_64	hesiod
hfsutils	3.2.6-7.2.2.ppc	-	-	hfsutils
htmlview	4.0.0-1.el5.noarch	4.0.0-1.el5.noarch	4.0.0-1.el5.noarch	htmlview
hwdata	0.194-1.noarch	0.194-1.noarch	0.194-1.noarch	hwdata
ifd-egate	0.05-15.ppc	-	0.05-15.x86_64	ifd-egate
imake	1.0.2-3.ppc	1.0.2-3.s390x	1.0.2-3.x86_64	imake
info	4.8-14.el5.ppc	4.8-14.el5.s390x	4.8-14.el5.x86_64	texinfo
initscripts	8.45.14.EL-1.ppc	8.45.14.EL-1.s390x	8.45.14.EL-1.x86_64	initscripts
iproute	2.6.18-4.el5.ppc	2.6.18-4.el5.s390x	2.6.18-4.el5.x86_64	iproute
iprutils	2.1.5-3.ppc	-	-	iprutils
ipsec-tools	0.6.5-8.el5.ppc	0.6.5-8.el5.s390x	0.6.5-8.el5.x86_64	ipsec-tools
iptables	1.3.5-1.2.1.ppc64	1.3.5-1.2.1.s390x	1.3.5-1.2.1.x86_64	iptables
iptables-ipv6	1.3.5-1.2.1.ppc64	1.3.5-1.2.1.s390x	1.3.5-1.2.1.x86_64	iptables
iptstate	1.4-1.1.2.2.ppc	1.4-1.1.2.2.s390x	1.4-1.1.2.2.x86_64	iptstate
iputils	20020927-43.el5.ppc	20020927-43.el5.s390x	20020927-43.el5.x86_64	iputils
irda-utils	0.9.17-2.fc6.ppc	-	0.9.17-2.fc6.x86_64	irda-utils
irqbalance	1.13-9.el5.ppc	-	1.13-9.el5.x86_64	irqbalance
irqbalance#2	1.13-9.el5.ppc64	-	-	irqbalance
jwhois	3.2.3-8.el5.ppc	3.2.3-8.el5.s390x	3.2.3-8.el5.x86_64	jwhois
kbd	1.12-19.el5.ppc	-	1.12-19.el5.x86_64	kbd
kernel	2.6.18-8.1.3.1spp.80.el5.ppc64	2.6.18-8.1.3.1spp.80.el5.s390x	2.6.18-8.1.3.1spp.80.el5.x86_64	kernel
kernel-devel	2.6.18-8.1.3.1spp.80.el5.ppc64	2.6.18-8.1.3.1spp.80.el5.s390x	2.6.18-8.1.3.1spp.80.el5.x86_64	kernel
kernel-headers	2.6.18-8.el5.ppc64	2.6.18-8.1.3.el5.s390x	2.6.18-8.el5.x86_64	kernel
keyutils-libs	1.2-1.el5.ppc	1.2-1.el5.s390x	1.2-1.el5.x86_64	keyutils
keyutils-libs-devel	1.2-1.el5.ppc	1.2-1.el5.s390	1.2-1.el5.i386	keyutils
keyutils-libs-devel#2	1.2-1.el5.ppc64	1.2-1.el5.s390x	1.2-1.el5.x86_64	keyutils
kpartx	0.4.7-8.el5.ppc	0.4.7-8.el5.s390x	0.4.7-8.el5.x86_64	device-mapper-multipath
krb5-devel	1.5-17.ppc64	1.5-17.s390x	1.5-17.x86_64	krb5
krb5-libs	1.5-17.ppc	1.5-17.s390	1.5-17.i386	krb5
krb5-libs#2	1.5-17.ppc64	1.5-17.s390x	1.5-17.x86_64	krb5
krb5-workstation	1.5-17.ppc	1.5-17.s390x	1.5-17.x86_64	krb5
ksh	20060214-1.4.ppc	20060214-1.4.s390x	20060214-1.4.x86_64	ksh
kudzu	1.2.57.1.13-1.ppc	1.2.57.1.13-1.s390x	1.2.57.1.13-1.x86_64	kudzu
less	394-5.el5.ppc	394-5.el5.s390x	394-5.el5.x86_64	less
lftp	3.5.1-2.fc6.ppc	3.5.1-2.fc6.s390x	3.5.1-2.fc6.x86_64	lftp
libFS	1.0.0-3.1.ppc	1.0.0-3.1.s390x	1.0.0-3.1.x86_64	libFS
libICE	1.0.1-2.1.ppc	1.0.1-2.1.s390	1.0.1-2.1.i386	libICE
libICE#2	1.0.1-2.1.ppc64	1.0.1-2.1.s390x	1.0.1-2.1.x86_64	libICE
libIDL	0.8.7-1.fc6.ppc	-	0.8.7-1.fc6.x86_64	libIDL
libSM	1.0.1-3.1.ppc	1.0.1-3.1.s390	1.0.1-3.1.i386	libSM
libSM#2	1.0.1-3.1.ppc64	1.0.1-3.1.s390x	1.0.1-3.1.x86_64	libSM
libX11	1.0.3-8.el5.ppc	1.0.3-8.el5.s390	1.0.3-8.el5.i386	libX11
libX11#2	1.0.3-8.el5.ppc64	1.0.3-8.el5.s390x	1.0.3-8.el5.x86_64	libX11
libXau	1.0.1-3.1.ppc	1.0.1-3.1.s390	1.0.1-3.1.i386	libXau
libXau#2	1.0.1-3.1.ppc64	1.0.1-3.1.s390x	1.0.1-3.1.x86_64	libXau
libXcursor	1.1.7-1.1.ppc	-	1.1.7-1.1.x86_64	libXcursor
libXdmcp	1.0.1-2.1.ppc	1.0.1-2.1.s390	1.0.1-2.1.i386	libXdmcp
libXdmcp#2	1.0.1-2.1.ppc64	1.0.1-2.1.s390x	1.0.1-2.1.x86_64	libXdmcp
libXext	1.0.1-2.1.ppc	1.0.1-2.1.s390	1.0.1-2.1.i386	libXext
libXext#2	1.0.1-2.1.ppc64	1.0.1-2.1.s390x	1.0.1-2.1.x86_64	libXext
libXfixes	4.0.1-2.1.ppc	-	4.0.1-2.1.x86_64	libXfixes
libXfont	1.2.2-1.fc6.ppc	1.2.2-1.fc6.s390x	1.2.2-1.fc6.x86_64	libXfont
libXft	2.1.10-1.1.ppc	2.1.10-1.1.s390x	2.1.10-1.1.x86_64	libXft
libXft#2	2.1.10-1.1.ppc64	-	-	libXft
libXi	1.0.1-3.1.ppc	1.0.1-3.1.s390	1.0.1-3.1.i386	libXi
libXi#2	1.0.1-3.1.ppc64	1.0.1-3.1.s390x	1.0.1-3.1.x86_64	libXi
libXinerama	1.0.1-2.1.ppc	-	1.0.1-2.1.x86_64	libXinerama
libXrandr	1.1.1-3.1.ppc	-	1.1.1-3.1.x86_64	libXrandr
libXrender	0.9.1-3.1.ppc	0.9.1-3.1.s390x	0.9.1-3.1.x86_64	libXrender
libXrender#2	0.9.1-3.1.ppc64	-	-	libXrender
libXres	1.0.1-3.1.ppc	-	1.0.1-3.1.x86_64	libXres
libXt	1.0.2-3.1.fc6.ppc	1.0.2-3.1.fc6.s390	1.0.2-3.1.fc6.i386	libXt
libXt#2	1.0.2-3.1.fc6.ppc64	1.0.2-3.1.fc6.s390x	1.0.2-3.1.fc6.x86_64	libXt
libXxf86vm	1.0.1-3.1.ppc	1.0.1-3.1.s390	1.0.1-3.1.i386	libXxf86vm
libXxf86vm#2	1.0.1-3.1.ppc64	1.0.1-3.1.s390x	1.0.1-3.1.x86_64	libXxf86vm
libacl	2.2.39-2.1.el5.ppc	2.2.39-2.1.el5.s390	2.2.39-2.1.el5.i386	acl
libacl#2	2.2.39-2.1.el5.ppc64	2.2.39-2.1.el5.s390x	2.2.39-2.1.el5.x86_64	acl
libacl-devel	2.2.39-2.1.el5.ppc	2.2.39-2.1.el5.s390	2.2.39-2.1.el5.i386	acl
libacl-devel#2	2.2.39-2.1.el5.ppc64	2.2.39-2.1.el5.s390x	2.2.39-2.1.el5.x86_64	acl
libaio	0.3.106-3.2.ppc	0.3.106-3.2.s390	0.3.106-3.2.i386	libaio
libaio#2	0.3.106-3.2.ppc64	0.3.106-3.2.s390x	0.3.106-3.2.x86_64	libaio
libattr	2.4.32-1.1.ppc	2.4.32-1.1.s390	2.4.32-1.1.i386	attr
libattr#2	2.4.32-1.1.ppc64	2.4.32-1.1.s390x	2.4.32-1.1.x86_64	attr
libattr-devel	2.4.32-1.1.ppc	2.4.32-1.1.s390	2.4.32-1.1.i386	attr
libattr-devel#2	2.4.32-1.1.ppc64	2.4.32-1.1.s390x	2.4.32-1.1.x86_64	attr
libcap	1.10-26.ppc	1.10-26.s390	1.10-26.i386	libcap
libcap#2	1.10-26.ppc64	1.10-26.s390x	1.10-26.x86_64	libcap
libcap-devel	1.10-26.ppc	1.10-26.s390	1.10-26.i386	libcap
libcap-devel#2	1.10-26.ppc64	1.10-26.s390x	1.10-26.x86_64	libcap
libdrm	2.0.2-1.1.ppc	-	2.0.2-1.1.i386	libdrm
libdrm#2	-	-	2.0.2-1.1.x86_64	libdrm
libevent	1.1a-3.2.1.ppc	1.1a-3.2.1.s390x	1.1a-3.2.1.x86_64	libevent
libevent#2	1.1a-3.2.1.ppc64	-	-	libevent
libfontenc	1.0.2-2.2.el5.ppc	1.0.2-2.2.el5.s390x	1.0.2-2.2.el5.x86_64	libfontenc
libfontenc#2	1.0.2-2.2.el5.ppc64	-	-	libfontenc
libgcc	4.1.1-52.el5.ppc	4.1.1-52.el5.s390	4.1.1-52.el5.i386	gcc
libgcc#2	4.1.1-52.el5.ppc64	4.1.1-52.el5.s390x	4.1.1-52.el5.x86_64	gcc
libgcrypt	1.2.3-1.ppc	1.2.3-1.s390	1.2.3-1.i386	libgcrypt
libgcrypt#2	1.2.3-1.ppc64	1.2.3-1.s390x	1.2.3-1.x86_64	libgcrypt
libgomp	4.1.1-52.el5.ppc	4.1.1-52.el5.s390x	4.1.1-52.el5.x86_64	gcc
libgpg-error	1.4-2.ppc	1.4-2.s390	1.4-2.i386	libgpg-error
libgpg-error#2	1.4-2.ppc64	1.4-2.s390x	1.4-2.x86_64	libgpg-error

libgssapi	0.10-2.ppc	0.10-2.s390x	0.10-2.x86_64	libgssapi
libhugetlbfs	1.0.1-1.el5.ppc	-	1.0.1-1.el5.x86_64	libhugetlbfs
libhugetlbfs-lib	1.0.1-1.el5.ppc64	-	1.0.1-1.el5.x86_64	libhugetlbfs
libidn	0.6.5-1.1.ppc	0.6.5-1.1.s390x	0.6.5-1.1.x86_64	libidn
libjpeg	6b-37.ppc	6b-37.s390	6b-37.i386	libjpeg
libjpeg#2	6b-37.ppc64	6b-37.s390x	6b-37.x86_64	libjpeg
libnl	1.0-0.10.pre5.4.ppc	1.0-0.10.pre5.4.s390x	1.0-0.10.pre5.4.x86_64	libnl
libnotify	0.4.2-6.el5.ppc	-	0.4.2-6.el5.x86_64	libnotify
libpcap	0.9.4-8.1.ppc	-	0.9.4-8.1.x86_64	tcpdump
libpng	1.2.10-7.ppc	1.2.10-7.s390	1.2.10-7.i386	libpng
libpng#2	1.2.10-7.ppc64	1.2.10-7.s390x	1.2.10-7.x86_64	libpng
librtas	1.2.4-3.el5.ppc	-	-	librtas
libselinux	1.33.4-4.el5.ppc	1.33.4-4.el5.s390	1.33.4-4.el5.i386	libselinux
libselinux#2	1.33.4-4.el5.ppc64	1.33.4-4.el5.s390x	1.33.4-4.el5.x86_64	libselinux
libselinux-devel	1.33.4-4.el5.ppc	1.33.4-4.el5.s390x	1.33.4-4.el5.x86_64	libselinux
libselinux-python	1.33.4-4.el5.ppc	1.33.4-4.el5.s390x	1.33.4-4.el5.x86_64	libselinux
libsemanage	1.9.1-3.el5.ppc	1.9.1-3.el5.s390x	1.9.1-3.el5.x86_64	libsemanage
libsemanage-devel	1.9.1-3.el5.ppc	1.9.1-3.el5.s390x	1.9.1-3.el5.x86_64	libsemanage
libsepol	1.15.2-1.el5.ppc	1.15.2-1.el5.s390	1.15.2-1.el5.i386	libsepol
libsepol#2	1.15.2-1.el5.ppc64	1.15.2-1.el5.s390x	1.15.2-1.el5.x86_64	libsepol
libsepol-devel	1.15.2-1.el5.ppc	1.15.2-1.el5.s390x	1.15.2-1.el5.x86_64	libsepol
libstdc++	4.1.1-52.el5.ppc	4.1.1-52.el5.s390	4.1.1-52.el5.i386	gcc
libstdc++#2	4.1.1-52.el5.ppc64	4.1.1-52.el5.s390x	4.1.1-52.el5.x86_64	gcc
libstdc++-devel	4.1.1-52.el5.ppc64	4.1.1-52.el5.s390x	4.1.1-52.el5.x86_64	gcc
libsysfs	2.0.0-6.ppc	2.0.0-6.s390x	2.0.0-6.x86_64	sysfsutils
libtermcap	2.0.8-46.1.ppc	2.0.8-46.1.s390	2.0.8-46.1.i386	libtermcap
libtermcap#2	2.0.8-46.1.ppc64	2.0.8-46.1.s390x	2.0.8-46.1.x86_64	libtermcap
libtermcap-devel	2.0.8-46.1.ppc64	2.0.8-46.1.s390x	2.0.8-46.1.x86_64	libtermcap
libtiff	3.8.2-7.el5.ppc	3.8.2-7.el5.s390	3.8.2-7.el5.i386	libtiff
libtiff#2	3.8.2-7.el5.ppc64	3.8.2-7.el5.s390x	3.8.2-7.el5.x86_64	libtiff
libusb	0.1.12-5.1.ppc	-	0.1.12-5.1.x86_64	libusb
libusb#2	0.1.12-5.1.ppc64	-	-	libusb
libuser	0.54.7-2.el5.1.ppc	0.54.7-2.el5.1.s390x	0.54.7-2.el5.1.x86_64	libuser
libuser-devel	0.54.7-2.el5.1.ppc	0.54.7-2.el5.1.s390x	0.54.7-2.el5.1.x86_64	libuser
libutempter	1.1.4-3.fc6.ppc	1.1.4-3.fc6.s390	1.1.4-3.fc6.i386	libutempter
libutempter#2	1.1.4-3.fc6.ppc64	1.1.4-3.fc6.s390x	1.1.4-3.fc6.x86_64	libutempter
libvolume_id	095-14.5.el5.ppc	095-14.5.el5.s390x	095-14.5.el5.x86_64	udev
libvolume_id#2	095-14.5.el5.ppc64	-	-	udev
libwnck	2.16.0-4.fc6.ppc	-	2.16.0-4.fc6.x86_64	libwnck
libxml2	2.6.26-2.1.2.ppc	2.6.26-2.1.2.s390x	2.6.26-2.1.2.x86_64	libxml2
libxml2-python	2.6.26-2.1.2.ppc	2.6.26-2.1.2.s390x	2.6.26-2.1.2.x86_64	libxml2
logrotate	3.7.4-7.ppc	3.7.4-7.s390x	3.7.4-7.x86_64	logrotate
logwatch	7.3-5.noarch	7.3-5.noarch	7.3-5.noarch	logwatch
lsof	4.78-3.ppc	4.78-3.s390x	4.78-3.x86_64	lsof
lspp-ea14-config-ibm	0.55-1.noarch	0.55-1.noarch	0.55-1.noarch	lspp-ea14-config-ibm
lvm2	2.02.16-3.el5.ppc	2.02.16-3.el5.s390x	2.02.16-3.el5.x86_64	lvm2
m2crypto	0.16-6.el5.1.ppc	0.16-6.el5.1.s390x	0.16-6.el5.1.x86_64	m2crypto
m4	1.4.5-3.el5.1.ppc	1.4.5-3.el5.1.s390x	1.4.5-3.el5.1.x86_64	m4
mailcap	2.1.23-1.fc6.noarch	2.1.23-1.fc6.noarch	2.1.23-1.fc6.noarch	mailcap
mailx	8.1.1-44.2.2.ppc	8.1.1-44.2.2.s390x	8.1.1-44.2.2.x86_64	mailx
make	3.81-1.1.ppc	3.81-1.1.s390x	3.81-1.1.x86_64	make
man	1.6d-1.1.ppc	1.6d-1.1.s390x	1.6d-1.1.x86_64	man
man-pages	2.39-9.el5.noarch	2.39-9.el5.noarch	2.39-9.el5.noarch	man-pages
mcelog	-	-	0.7-1.22.fc6.x86_64	mcelog
mcstrans	0.2.3-1.el5.ppc	0.2.3-1.el5.s390x	0.2.3-1.el5.x86_64	mcstrans
mdadm	2.5.4-3.el5.ppc	2.5.4-3.el5.s390x	2.5.4-3.el5.x86_64	mdadm
mesa-libGL	6.5.1-7.2.el5.ppc	6.5.1-7.2.el5.s390	6.5.1-7.2.el5.i386	mesa
mesa-libGL#2	6.5.1-7.2.el5.ppc64	6.5.1-7.2.el5.s390x	6.5.1-7.2.el5.x86_64	mesa
mgetty	1.1.33-9.fc6.ppc	1.1.33-9.fc6.s390x	1.1.33-9.fc6.x86_64	mgetty
microcode_ctl	-	-	1.15-1.40.el5.x86_64	microcode_ctl
mingetty	1.07-5.2.2.ppc	1.07-5.2.2.s390x	1.07-5.2.2.x86_64	mingetty
mkbootdisk	-	-	1.5.3-2.1.x86_64	mkbootdisk
mkinitrd	5.1.19.6-1.ppc	5.1.19.6-1.s390	5.1.19.6-1.i386	mkinitrd
mkinitrd#2	5.1.19.6-1.ppc64	5.1.19.6-1.s390x	5.1.19.6-1.x86_64	mkinitrd
mktemp	1.5-23.2.2.ppc	1.5-23.2.2.s390x	1.5-23.2.2.x86_64	mktemp
mlocate	0.15-1.el5.ppc	0.15-1.el5.s390x	0.15-1.el5.x86_64	mlocate
module-init-tools	3.3-0.pre3.1.16.el5.ppc	3.3-0.pre3.1.16.el5.s390x	3.3-0.pre3.1.16.el5.x86_64	module-init-tools
mtools	3.9.10-2.fc6.ppc	3.9.10-2.fc6.s390x	3.9.10-2.fc6.x86_64	mtools
mtr	0.71-3.1.ppc	0.71-3.1.s390x	0.71-3.1.x86_64	mtr
nano	1.3.12-1.1.ppc	1.3.12-1.1.s390x	1.3.12-1.1.x86_64	nano
nash	5.1.19.6-1.ppc	5.1.19.6-1.s390x	5.1.19.6-1.x86_64	mkinitrd
nc	1.84-10.fc6.ppc	1.84-10.fc6.s390x	1.84-10.fc6.x86_64	nc
ncurses	5.5-24.20060715.ppc	5.5-24.20060715.s390	5.5-24.20060715.i386	ncurses
ncurses#2	5.5-24.20060715.ppc64	5.5-24.20060715.s390x	5.5-24.20060715.x86_64	ncurses
net-snmp-libs	5.3.1-14.el5.ppc	5.3.1-14.el5.s390x	5.3.1-14.el5.x86_64	net-snmp
net-tools	1.60-73.ppc	1.60-73.s390x	1.60-73.x86_64	net-tools
netlabel_tools	0.17-9.el5.ppc	0.17-9.el5.s390x	0.17-9.el5.x86_64	netlabel_tools
newt	0.52.2-9.ppc	0.52.2-9.s390x	0.52.2-9.x86_64	newt
newt#2	0.52.2-9.ppc64	-	-	newt
nfs-utils	1.0.9-16.el5.ppc	1.0.9-16.el5.s390x	1.0.9-16.el5.x86_64	nfs-utils
nfs-utils-lib	1.0.8-7.2.ppc	1.0.8-7.2.s390x	1.0.8-7.2.x86_64	nfs-utils-lib
notification-daemon	0.3.5-8.el5.ppc	-	0.3.5-8.el5.x86_64	notification-daemon
nscd	2.5-12.ppc	2.5-12.s390x	2.5-12.x86_64	glibc
nspr	4.6.5-1.el5.ppc	4.6.5-1.el5.s390	4.6.5-1.el5.i386	nspr
nspr#2	4.6.5-1.el5.ppc64	4.6.5-1.el5.s390x	4.6.5-1.el5.x86_64	nspr
nss	3.11.5-1.el5.ppc	3.11.5-1.el5.s390	3.11.5-1.el5.i386	nss
nss#2	3.11.5-1.el5.ppc64	3.11.5-1.el5.s390x	3.11.5-1.el5.x86_64	nss
nss-tools	3.11.5-1.el5.ppc	-	3.11.5-1.el5.x86_64	nss
nss_db	2.2-35.1.ppc	2.2-35.1.s390	2.2-35.1.i386	nss_db
nss_db#2	2.2-35.1.ppc64	2.2-35.1.s390x	2.2-35.1.x86_64	nss_db
nss_ldap	253-3.ppc	253-3.s390	253-3.i386	nss_ldap
nss_ldap#2	253-3.ppc64	253-3.s390x	253-3.x86_64	nss_ldap
ntsysv	1.3.30.1-1.ppc	1.3.30.1-1.s390x	1.3.30.1-1.x86_64	chkconfig
numactl	0.9.8-2.el5.ppc	-	0.9.8-2.el5.i386	numactl
numactl#2	0.9.8-2.el5.ppc64	-	0.9.8-2.el5.x86_64	numactl
openldap	2.3.27-5.ppc	2.3.27-5.s390	2.3.27-5.i386	openldap
openldap#2	2.3.27-5.ppc64	2.3.27-5.s390x	2.3.27-5.x86_64	openldap
openssh	4.3p2-21.el5.ppc	4.3p2-21.el5.s390x	4.3p2-21.el5.x86_64	openssh
openssh-clients	4.3p2-21.el5.ppc	4.3p2-21.el5.s390x	4.3p2-21.el5.x86_64	openssh
openssh-server	4.3p2-21.el5.ppc	4.3p2-21.el5.s390x	4.3p2-21.el5.x86_64	openssh
openssl	0.9.8b-8.3.el5.ppc	0.9.8b-8.3.el5.s390	0.9.8b-8.3.el5.i686	openssl
openssl#2	0.9.8b-8.3.el5.ppc64	0.9.8b-8.3.el5.s390x	0.9.8b-8.3.el5.x86_64	openssl
openssl-devel	0.9.8b-8.3.el5.ppc	0.9.8b-8.3.el5.s390x	0.9.8b-8.3.el5.x86_64	openssl
pam	0.99.6.2-3.22.el5.ppc	0.99.6.2-3.22.el5.s390	0.99.6.2-3.22.el5.i386	pam
pam#2	0.99.6.2-3.22.el5.ppc64	0.99.6.2-3.22.el5.s390x	0.99.6.2-3.22.el5.x86_64	pam
pam-devel	0.99.6.2-3.22.el5.ppc	0.99.6.2-3.22.el5.s390x	0.99.6.2-3.22.el5.x86_64	pam
pam_ccreds	3-5.ppc	3-5.s390	3-5.i386	pam_ccreds
pam_ccreds#2	3-5.ppc64	3-5.s390x	3-5.x86_64	pam_ccreds
pam_krb5	2.2.11-1.ppc	2.2.11-1.s390	2.2.11-1.i386	pam_krb5
pam_krb5#2	2.2.11-1.ppc64	2.2.11-1.s390x	2.2.11-1.x86_64	pam_krb5
pam_passwdqc	1.0.2-1.2.2.ppc	1.0.2-1.2.2.s390	1.0.2-1.2.2.i386	pam_passwdqc
pam_passwdqc#2	1.0.2-1.2.2.ppc64	1.0.2-1.2.2.s390x	1.0.2-1.2.2.x86_64	pam_passwdqc
pam_pkcs11	0.5.3-23.ppc	0.5.3-23.s390	0.5.3-23.i386	pam_pkcs11
pam_pkcs11#2	0.5.3-23.ppc64	0.5.3-23.s390x	0.5.3-23.x86_64	pam_pkcs11
pam_smb	1.1.7-7.2.1.ppc	1.1.7-7.2.1.s390	1.1.7-7.2.1.i386	pam_smb
pam_smb#2	1.1.7-7.2.1.ppc64	1.1.7-7.2.1.s390x	1.1.7-7.2.1.x86_64	pam_smb
pango	1.14.9-3.el5.ppc	1.14.9-3.el5.s390x	1.14.9-3.el5.x86_64	pango
paps	0.6.6-17.el5.ppc	0.6.6-17.el5.s390x	0.6.6-17.el5.x86_64	paps
parted	1.8.1-4.el5.ppc	1.8.1-4.el5.s390	1.8.1-4.el5.i386	parted
parted#2	1.8.1-4.el5.ppc64	1.8.1-4.el5.s390x	1.8.1-4.el5.x86_64	parted
passwd	0.73-1.ppc	0.73-1.s390x	0.73-1.x86_64	passwd
patch	2.5.4-29.2.2.ppc	2.5.4-29.2.2.s390x	2.5.4-29.2.2.x86_64	patch
pax	3.4-1.2.2.ppc	3.4-1.2.2.s390x	3.4-1.2.2.x86_64	pax
pciutils	2.2.3-4.ppc	2.2.3-4.s390x	2.2.3-4.x86_64	pciutils
pciutils-devel	2.2.3-4.ppc	2.2.3-4.s390	2.2.3-4.i386	pciutils
pciutils-devel#2	2.2.3-4.ppc64	2.2.3-4.s390x	2.2.3-4.x86_64	pciutils
pcmciautils	014-5.ppc	-	014-5.x86_64	pcmciautils
pcre	6.6-1.1.ppc	6.6-1.1.s390x	6.6-1.1.x86_64	pcre
pcsc-lite	1.3.1-7.ppc	-	1.3.1-7.x86_64	pcsc-lite
pcsc-lite-libs	1.3.1-7.ppc	-	1.3.1-7.x86_64	pcsc-lite
perl	5.8.8-10.ppc	5.8.8-10.s390x	5.8.8-10.x86_64	perl
perl-Digest-HMAC	1.01-15.noarch	1.01-15.noarch	1.01-15.noarch	perl-Digest-HMAC
perl-Digest-SHA1	2.11-1.2.1.ppc	2.11-1.2.1.s390x	2.11-1.2.1.x86_64	perl-Digest-SHA1
perl-String-CRC32	1.4-2.fc6.ppc	1.4-2.fc6.s390x	1.4-2.fc6.x86_64	perl-String-CRC32

pinfo	0.6.9-1.fc6.ppc	0.6.9-1.fc6.s390x	0.6.9-1.fc6.x86_64	pinfo
pkgconfig	0.21-1.fc6.ppc	0.21-1.fc6.s390x	0.21-1.fc6.x86_64	pkgconfig
pkinit-nss	0.3.5-1.el5.ppc	0.3.5-1.el5.s390x	0.3.5-1.el5.x86_64	pkinit-nss
pm-utils	0.19-3.ppc	-	0.19-3.x86_64	pm-utils
policycoreutils	1.33.12-8.el5.ppc	1.33.12-8.el5.s390x	1.33.12-8.el5.x86_64	policycoreutils
policycoreutils-newrole	1.33.12-8.el5.ppc	1.33.12-8.el5.s390x	1.33.12-8.el5.x86_64	policycoreutils
popt	1.10.2-37.el5.ppc	1.10.2-37.el5.s390x	1.10.2-37.el5.x86_64	rpm
popt#2	1.10.2-37.el5.ppc64	-	-	rpm
portmap	4.0-65.2.2.1.ppc	4.0-65.2.2.1.s390x	4.0-65.2.2.1.x86_64	portmap
postfix	2.3.3-2.ppc	2.3.3-2.s390x	2.3.3-2.x86_64	postfix
ppc64-utils	0.11-2.ppc	-	-	ppc64-utils
ppp	2.4.4-1.el5.ppc	-	2.4.4-1.el5.x86_64	ppp
prelink	0.3.9-2.ppc	0.3.9-2.s390x	0.3.9-2.x86_64	prelink
procps	3.2.7-8.1.el5.ppc	3.2.7-8.1.el5.s390x	3.2.7-8.1.el5.x86_64	procps
procps#2	3.2.7-8.1.el5.ppc64	-	-	procps
psacct	6.3.2-41.1.ppc	6.3.2-41.1.s390x	6.3.2-41.1.x86_64	psacct
psmisc	22.2-5.ppc	22.2-5.s390x	22.2-5.x86_64	psmisc
pyOpenSSL	0.6-1.p24.7.2.2.ppc	0.6-1.p24.7.2.2.s390x	0.6-1.p24.7.2.2.x86_64	pyOpenSSL
pygobject2	2.12.1-5.el5.ppc	2.12.1-5.el5.s390x	2.12.1-5.el5.x86_64	pygobject2
python	2.4.3-19.el5.ppc	2.4.3-19.el5.s390x	2.4.3-19.el5.x86_64	python
python-devel	2.4.3-19.el5.ppc	2.4.3-19.el5.s390	2.4.3-19.el5.i386	python
python-devel#2	2.4.3-19.el5.ppc64	2.4.3-19.el5.s390x	2.4.3-19.el5.x86_64	python
python-elementtree	1.2.6-5.ppc	1.2.6-5.s390x	1.2.6-5.x86_64	python-elementtree
python-sqlite	1.1.7-1.2.1.ppc	1.1.7-1.2.1.s390x	1.1.7-1.2.1.x86_64	python-sqlite
python-urlgrabber	3.1.0-2.noarch	3.1.0-2.noarch	3.1.0-2.noarch	python-urlgrabber
quota	3.13-1.2.3.2.el5.ppc	3.13-1.2.3.2.el5.s390x	3.13-1.2.3.2.el5.x86_64	quota
rdate	1.4-6.ppc	1.4-6.s390x	1.4-6.x86_64	rdate
rdist	6.1.5-44.ppc	6.1.5-44.s390x	6.1.5-44.x86_64	rdist
readahead	1.3-7.el5.ppc	1.3-7.el5.s390x	1.3-7.el5.x86_64	readahead
readline	5.1-1.1.ppc	5.1-1.1.s390	5.1-1.1.i386	readline
readline#2	5.1-1.1.ppc64	5.1-1.1.s390x	5.1-1.1.x86_64	readline
readline-devel	5.1-1.1.ppc	5.1-1.1.s390	5.1-1.1.i386	readline
readline-devel#2	5.1-1.1.ppc64	5.1-1.1.s390x	5.1-1.1.x86_64	readline
redhat-logos	4.9.16-1.noarch	4.9.16-1.noarch	4.9.16-1.noarch	redhat-logos
redhat-lsb	3.1-12.2.EL.ppc	3.1-12.2.EL.s390	3.1-12.2.EL.i386	redhat-lsb
redhat-lsb#2	3.1-12.2.EL.ppc64	3.1-12.2.EL.s390x	3.1-12.2.EL.x86_64	redhat-lsb
redhat-menus	6.7.8-1.el5.noarch	6.7.8-1.el5.noarch	6.7.8-1.el5.noarch	redhat-menus
redhat-release	5Server-5.0.0.9.ppc	5Server-5.0.0.9.s390x	5Server-5.0.0.9.x86_64	redhat-release
redhat-release-notes	5Server-5.ppc	5Server-5.s390x	5Server-5.x86_64	redhat-release-notes
rhel-instnum	1.0.7-1.el5.noarch	1.0.7-1.el5.noarch	1.0.7-1.el5.noarch	rhel-instnum
rhn-check	0.4.13-1.el5.noarch	0.4.13-1.el5.noarch	0.4.13-1.el5.noarch	rhn-client-tools
rhn-client-tools	0.4.13-1.el5.noarch	0.4.13-1.el5.noarch	0.4.13-1.el5.noarch	rhn-client-tools
rhn-setup	0.4.13-1.el5.noarch	0.4.13-1.el5.noarch	0.4.13-1.el5.noarch	rhn-client-tools
rhnlib	2.2.5-1.el5.noarch	2.2.5-1.el5.noarch	2.2.5-1.el5.noarch	rhnlib
rhnsd	4.6.1-1.el5.ppc	4.6.1-1.el5.s390x	4.6.1-1.el5.x86_64	rhnsd
rhpl	0.194.1-1.ppc	0.194.1-1.s390x	0.194.1-1.x86_64	rhpl
rmt	0.4b41-2.fc6.ppc	0.4b41-2.fc6.s390x	0.4b41-2.fc6.x86_64	dump
rng-utils	2.0-1.14.1.fc6.ppc	2.0-1.14.1.fc6.s390x	2.0-1.14.1.fc6.x86_64	rng-utils
rootfiles	8.1-1.1.1.noarch	8.1-1.1.1.noarch	8.1-1.1.1.noarch	rootfiles
rp-pppoe	3.5-32.1.ppc	-	3.5-32.1.x86_64	rp-pppoe
rpm	4.4.2-37.el5.ppc	4.4.2-37.el5.s390x	4.4.2-37.el5.x86_64	rpm
rpm-build	4.4.2-37.el5.ppc	4.4.2-37.el5.s390x	4.4.2-37.el5.x86_64	rpm
rpm-libs	4.4.2-37.el5.ppc	4.4.2-37.el5.s390x	4.4.2-37.el5.x86_64	rpm
rpm-python	4.4.2-37.el5.ppc	4.4.2-37.el5.s390x	4.4.2-37.el5.x86_64	rpm
rsh	0.17-37.el5.ppc	0.17-37.el5.s390x	0.17-37.el5.x86_64	rsh
rsync	2.6.8-3.1.ppc	2.6.8-3.1.s390x	2.6.8-3.1.x86_64	rsync
s390utils	-	1.5.3-10.el5.6.s390x	-	s390utils
sed	4.1.5-5.fc6.ppc	4.1.5-5.fc6.s390x	4.1.5-5.fc6.x86_64	sed
selinux-policy	2.4.6-67.el5.noarch	2.4.6-67.el5.noarch	2.4.6-67.el5.noarch	selinux-policy
selinux-policy-devel	2.4.6-67.el5.noarch	2.4.6-67.el5.noarch	2.4.6-67.el5.noarch	selinux-policy
selinux-policy-mls	2.4.6-67.el5.noarch	2.4.6-67.el5.noarch	2.4.6-67.el5.noarch	selinux-policy
selinux-policy-strict	2.4.6-67.el5.noarch	2.4.6-67.el5.noarch	2.4.6-67.el5.noarch	selinux-policy
selinux-policy-targeted	2.4.6-67.el5.noarch	2.4.6-67.el5.noarch	2.4.6-67.el5.noarch	selinux-policy
setarch	2.0-1.1.ppc	2.0-1.1.s390x	2.0-1.1.x86_64	setarch
setools	3.0-3.el5.ppc	3.0-3.el5.s390x	3.0-3.el5.x86_64	setools
setserial	2.17-19.2.2.ppc	-	2.17-19.2.2.x86_64	setserial
setup	2.5.58-1.el5.noarch	2.5.58-1.el5.noarch	2.5.58-1.el5.noarch	setup
setuptool	1.19.2-1.ppc	1.19.2-1.s390x	1.19.2-1.x86_64	setuptool
shadow-utils	4.0.17-12.el5.ppc	4.0.17-12.el5.s390x	4.0.17-12.el5.x86_64	shadow-utils
slang	2.0.6-4.el5.ppc	2.0.6-4.el5.s390x	2.0.6-4.el5.x86_64	slang
slang#2	2.0.6-4.el5.ppc64	-	-	slang
smartmontools	5.36-3.1.el5.ppc	-	5.36-3.1.el5.x86_64	smartmontools
sos	1.3-1.el5.noarch	1.3-1.el5.noarch	1.3-1.el5.noarch	sos
specspo	13-1.el5.noarch	13-1.el5.noarch	13-1.el5.noarch	specspo
sqlite	3.3.6-2.ppc	3.3.6-2.s390x	3.3.6-2.x86_64	sqlite
star	1.5a75-1.ppc	1.5a75-1.s390x	1.5a75-1.x86_64	star
startup-notification	0.8-4.1.ppc	-	0.8-4.1.x86_64	startup-notification
startup-notification#2	0.8-4.1.ppc64	-	-	startup-notification
strace	4.5.15-1.el5.ppc	4.5.15-1.el5.s390x	4.5.15-1.el5.x86_64	strace
stunnel	4.15-2.ppc	4.15-2.s390x	4.15-2.x86_64	stunnel
sudo	1.6.8p12-10.ppc	1.6.8p12-10.s390x	1.6.8p12-10.x86_64	sudo
swig	1.3.29-2.el5.ppc	1.3.29-2.el5.s390x	1.3.29-2.el5.x86_64	swig
symlinks	1.2-24.2.2.ppc	1.2-24.2.2.s390x	1.2-24.2.2.x86_64	symlinks
sysfsutils	2.0.0-6.ppc	2.0.0-6.s390x	2.0.0-6.x86_64	sysfsutils
sysklogd	1.4.1-39.2.ppc	1.4.1-39.2.s390x	1.4.1-39.2.x86_64	sysklogd
syslinux	-	-	3.11-4.x86_64	syslinux
sysreport	1.4.3-10.el5.noarch	1.4.3-10.el5.noarch	1.4.3-10.el5.noarch	sysreport
system-config-network-tui	1.3.99-1.el5.noarch	1.3.99-1.el5.noarch	1.3.99-1.el5.noarch	system-config-network
system-config-securitylevel-tui	1.6.29.1-1.el5.ppc	1.6.29.1-1.el5.s390x	1.6.29.1-1.el5.x86_64	system-config-securitylevel
talk	0.17-29.2.2.ppc	0.17-29.2.2.s390x	0.17-29.2.2.x86_64	talk
tar	1.15.1-23.el5.ppc	1.15.1-23.el5.s390x	1.15.1-23.el5.x86_64	tar
tcl	8.4.13-3.fc6.ppc	8.4.13-3.fc6.s390	8.4.13-3.fc6.i386	tcl
tcl#2	8.4.13-3.fc6.ppc64	8.4.13-3.fc6.s390x	8.4.13-3.fc6.x86_64	tcl
tcp_wrappers	7.6-40.2.1.ppc	7.6-40.2.1.s390	7.6-40.2.1.i386	tcp_wrappers
tcp_wrappers#2	7.6-40.2.1.ppc64	7.6-40.2.1.s390x	7.6-40.2.1.x86_64	tcp_wrappers
tcpdump	3.9.4-8.1.ppc	3.9.4-8.1.s390x	3.9.4-8.1.x86_64	tcpdump
tcsh	6.14-12.el5.ppc	6.14-12.el5.s390x	6.14-12.el5.x86_64	tcsh
telnet	0.17-38.el5.ppc	0.17-38.el5.s390x	0.17-38.el5.x86_64	telnet
termcap	5.5-1.20060701.1.noarch	5.5-1.20060701.1.noarch	5.5-1.20060701.1.noarch	termcap
texinfo	4.8-14.el5.ppc	4.8-14.el5.s390x	4.8-14.el5.x86_64	texinfo
time	1.7-27.2.2.ppc	1.7-27.2.2.s390x	1.7-27.2.2.x86_64	time
tk	8.4.13-3.fc6.ppc	8.4.13-3.fc6.s390	8.4.13-3.fc6.i386	tk
tk#2	8.4.13-3.fc6.ppc64	8.4.13-3.fc6.s390x	8.4.13-3.fc6.x86_64	tk
tmpwatch	2.9.7-1.1.el5.1.ppc	2.9.7-1.1.el5.1.s390x	2.9.7-1.1.el5.1.x86_64	tmpwatch
traceroute	2.0.1-2.el5.ppc	2.0.1-2.el5.s390x	2.0.1-2.el5.x86_64	traceroute
tree	1.5.0-4.ppc	1.5.0-4.s390x	1.5.0-4.x86_64	tree
ttmkfdir	3.0.9-23.el5.ppc	3.0.9-23.el5.s390x	3.0.9-23.el5.x86_64	ttmkfdir
tzdata	2006m-2.fc6.noarch	2006m-2.fc6.noarch	2006m-2.fc6.noarch	tzdata
udev	095-14.5.el5.ppc	095-14.5.el5.s390x	095-14.5.el5.x86_64	udev
unix2dos	2.2-26.2.2.ppc	2.2-26.2.2.s390x	2.2-26.2.2.x86_64	unix2dos
unzip	5.52-2.2.1.ppc	5.52-2.2.1.s390x	5.52-2.2.1.x86_64	unzip
urw-fonts	2.3-6.1.1.noarch	2.3-6.1.1.noarch	2.3-6.1.1.noarch	urw-fonts
usbutils	0.71-2.1.ppc	-	0.71-2.1.x86_64	usbutils
usermode	1.88-3.el5.ppc	1.88-3.el5.s390x	1.88-3.el5.x86_64	usermode
util-linux	2.13-0.44.el5.ppc	2.13-0.44.el5.s390x	2.13-0.44.el5.x86_64	util-linux
vconfig	1.9-2.1.ppc	1.9-2.1.s390x	1.9-2.1.x86_64	vconfig
vim-minimal	7.0.109-3.ppc	7.0.109-3.s390x	7.0.109-3.x86_64	vim
vixie-cron	4.1-68.el5.ppc	4.1-68.el5.s390x	4.1-68.el5.x86_64	vixie-cron
vsftpd	2.0.5-10.el5.ppc	2.0.5-10.el5.s390x	2.0.5-10.el5.x86_64	vsftpd
wget	1.10.2-7.el5.ppc	1.10.2-7.el5.s390x	1.10.2-7.el5.x86_64	wget
which	2.16-7.ppc	2.16-7.s390x	2.16-7.x86_64	which
wireless-tools	28-2.el5.ppc	-	28-2.el5.i386	wireless-tools
wireless-tools#2	28-2.el5.ppc64	-	28-2.el5.x86_64	wireless-tools
words	3.0-9.noarch	3.0-9.noarch	3.0-9.noarch	words
wpa_supplicant	0.4.8-10.1.fc6.ppc	-	0.4.8-10.1.fc6.x86_64	wpa_supplicant
xinetd	2.3.14-10.el5.ppc	2.3.14-10.el5.s390x	2.3.14-10.el5.x86_64	xinetd
xorg-x11-filesystem	7.1-2.fc6.noarch	7.1-2.fc6.noarch	7.1-2.fc6.noarch	xorg-x11-filesystem
xorg-x11-font-utils	7.1-2.ppc	7.1-2.s390x	7.1-2.x86_64	xorg-x11-font-utils
xorg-x11-xfs	1.0.2-3.1.ppc	1.0.2-3.1.s390x	1.0.2-3.1.x86_64	xorg-x11-xfs
yaboot	1.3.13-3.el5.ppc	-	-	yaboot
yp-tools	2.9-0.1.ppc	2.9-0.1.s390x	2.9-0.1.x86_64	yp-tools
ypbind	1.19-7.el5.ppc	1.19-7.el5.s390x	1.19-7.el5.x86_64	ypbind
yum	3.0.1-5.el5.noarch	3.0.1-5.el5.noarch	3.0.1-5.el5.noarch	yum
yum-metadata-parser	1.0-8.fc6.ppc	1.0-8.fc6.s390x	1.0-8.fc6.x86_64	yum-metadata-parser

yum-rhn-plugin	0.4.3-1.el5.noarch	0.4.3-1.el5.noarch	0.4.3-1.el5.noarch	yum-rhn-plugin
yum-updatesd	3.0.1-5.el5.noarch	3.0.1-5.el5.noarch	3.0.1-5.el5.noarch	yum
zip	2.31-1.2.2.ppc	2.31-1.2.2.s390x	2.31-1.2.2.x86_64	zip
zlib	1.2.3-3.ppc	1.2.3-3.s390	1.2.3-3.i386	zlib
zlib#2	1.2.3-3.ppc64	1.2.3-3.s390x	1.2.3-3.x86_64	zlib
zlib-devel	1.2.3-3.ppc	1.2.3-3.s390	1.2.3-3.i386	zlib
zlib-devel#2	1.2.3-3.ppc64	1.2.3-3.s390x	1.2.3-3.x86_64	zlib

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

- The "#2" suffix indicates that multiple copies of a package are installed using different word sizes. The suffix is not part of the package name.

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, both are also supported in the evaluated configuration.
- Both installation from CD 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 TOE and afforded the same physical protection as the TOE.
- The TOE supports two modes of operation: CAPP-mode and LSPP-mode. The software configuration for both modes is identical and the only difference is within the SELinux security module and the policy file for this module.

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

2.4.1 File systems

The following file system types are supported:

- the ext3 journaling filesystem,
- the read-only ISO 9660 filesystem for CD-ROM drives and DVD drives,
- The process file system, procfs (/proc) represents processes / tasks 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. 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 device driver specific information. Access to objects there can be restricted using the DAC mechanism (which consists of the permission bits only).
- The temporary filesystem (tmpfs) used as a fast nonpersistent 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) provides the SELinux policy API for userspace programs and is used for configuring the selinux system.
- The Hypervisor file system (hypfs) provides information about the hypervisor on System z systems.

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:

- System x: x3550 (rack mount), HS20 and HS21 (blades)

- Opteron (AMD): x3455 (rack mount), LS21 (blade)
- System p: any POWER5/POWER5+ compliant system or software
- System z: any z/Architecture compliant system or software

The following 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). **Note:** Serial devices are not supported on System z.
- printers compatible with PostScript level 1 or PCL 4 attached via parallel port or USB. Network printers are supported in CAPP mode only.
- all storage devices and backup devices supported by the TOE software (hard disks, CDROM drives, streamer drives, floppy disk drives) (except hot pluggable devices connected via USB or IEEE 1394 (Firewire) interfaces).
- all Ethernet and Token-Ring network adapters supported by the TOE software.

Note: peripheral devices are part of the TOE environment.

Note: the peripherals are physical peripherals for the System x, System p5 and eServer models. In the case of System z the TOE is executing within a logical partition and the peripherals used may be virtualized. The logical partitioning software is part of the "abstract machine" and therefore part of the TOE. The Evaluated Configuration Guide provides the required guidance on how to set up and configure the logical partitioning software and how to define the logical peripheral devices such that the TOE software operates securely in the logical partitioning environment on the System z platforms. Separate hardware management consoles used to configure the partitioning software are considered part of the TOE environment

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

2.4.3 TOE Environment

Several TOE systems may be interlinked in a network, and individual networks may be joined by bridges and/or routers, or by TOE systems which act as routers and/or gateways. Each of the TOE systems 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 systems 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.

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.

This Security Targets combines the threats, organizational policies and assumptions from [CAPP], [LSPP] and [RBACPP]. Those mentioned in [LSPP] are a superset of the ones mentioned in [CAPP]. In many cases [LSPP] and [RBACPP] have very similar threats, policies and assumptions, which the author of this Security Target has attempted to combine in a useful way.

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, including data in transit between systems.

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 security vulnerabilities that might be exploited in the intended environment for the TOE with medium level of expertise and effort. 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

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

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.

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 TOE complies with the following organizational security policies:

P.ACCESS (LSPP 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.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 or (in LSPP-mode) by the sensitivity label of the subject and object.

Application Note: Being able to model an organization's access control policy based on the three properties above ensures that the organization's policy can be mapped to the TOE with the security functions provided by the TOE. For example an access control policy based on time dependent or content dependent rules would not satisfy the above mentioned policy.

P.ACCOUNTABILITY

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

P.CLASSIFICATION (LSPP-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.

The method for classification of information is made based on criteria set forth by the organization. This is usually done on a basis of relative value to the organization and its interest to limit 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 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. The assumptions have been derived from [CAPP]/[LSPP] and [RBACPP]. In some cases those protection profiles have similar but not identical assumptions. Where possible this Security Target has combined those in a way that addresses the assumptions of all those protection profiles.

3.4.1 Physical Aspects

A.ASSET It is assumed that the value of the stored assets merits moderately intensive penetration or masquerading attacks.

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 including unauthorized modifications by potentially hostile outsiders.

3.4.2 Personnel Aspects

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

A.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 Users are trained to use the security functionality provided by the system appropriately.

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

3.4.3 Procedural Aspects (LSPP-mode only)

A.CLEARANCE Procedures exist for granting users authorization for access to specific security levels.

A.SENSITIVITY 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 systems with which the TOE communicates are assumed to be under the same management control and operate under the same security policy constraints. When operating in LSPP mode any data exported from the TOE to another system either with its sensitivity label or without the sensitivity label (over a single level connection) is assumed to be handled in accordance with its sensitivity label on any system that imports this data.

A.CONNECT All connections to peripheral devices and all network connections not using the secured protocols SSH v2.0 or SSL v3 reside within the controlled access facilities. When using labeled networking in LSPP mode, all network connections need to reside within the controlled access facilities because the secured protocols SSH and SSL do not protect the label information. 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

The security objectives have been derived from [CAPP]/[LSPP] and [RBACPP]. Where the protection profiles define similar but not identical security objectives this Security Target has attempted to combine them in a way that addresses the details of the security objectives of all source protection profiles.

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 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 and security relevant events. The TSF must present this information to authorized administrators. The information recorded with security relevant events must be in sufficient detail to help an administrator of the TOE detect attempted security violations or potential misconfiguration of the TOE security features that would leave the IT assets open to compromise.

O.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. Those functions must enable an authorized administrator to effectively manage the TOE and its security functions.

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 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 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 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, configured and administered in a secure manner. This includes the definition and assignment of roles.

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 and MAC 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 (e.g. IBM 3151 terminals) are used.

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 that are 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 [LSPP] and [RBACPP], tailored as described in section 7.2 of this document, and including some TOE specific extensions. Where appropriate and possible the author of this Security Target has combined security functional requirements included in [LSPP] and [RBACPP] into a single instantiation of the security functional requirement. In cases where this was not possible, the author included multiple instantiations of the individual security functional requirements to achieve compatibility with both [LSPP] and [RBACPP].

Security functional requirements referring to Mandatory Access Control (MAC, including references to sensitivity labels and user clearances) or Role-Based Access Control (RBAC) apply only when the TOE is operating in LSPP mode.

In CAPP mode, all roles specified in SFRs are subsumed within the single administrator role (root user with UID 0).

All instantiations are marked in **bold** within each of the requirements regardless if they have already been defined as instantiations in one of the Protection Profiles or not. Refinements are marked in ***bold and italics***. The reader should also be aware that where security functional requirements were defined in both [LSPP] and [RBACPP] the author of this ST has tried to combine them into one instantiation of the SFR were possible. If this was not possible, the SFR has been instantiated multiple times to address the requirements of both [LSPP] and [RBACPP]

Security Functional requirements in addition to those taken from [LSPP] and [RBACPP] are shown in green with TOE specific instantiations marked in **green and bold.**

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 the start-up and shutdown of the audit functions, and all auditable events for the basic level of audit except FIA_UID.2's user identity during failures. This includes also the:**

i. Assignment of Users, Roles and Privileges to Roles

ii. Deletion of Users, Roles and Privileges from Roles

iii. Creation and Deletion of Roles

FAU_GEN.1.2 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) **In LSPP mode, the sensitivity labels of subjects, objects, or information involved; and**

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

d) **For each audit event type, based on the auditable event definitions of the functional components included in this 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.**

Table 5-1: Auditable Events

Component	Event	Details (Event Names)
FAU_GEN.1	Start-up and shutdown of the audit functions.	Events ”auditd start”, ”auditd halt” (from `auditd`)
FAU_GEN.2	None	
FAU_SAR.1	Reading of information from the audit records.	Syscall *open* (on the audit log files)
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.	Event “AUDIT_CONFIG_CHANGE”; syscalls *open*, *link*, *unlink*, *rename*, *truncate* (write access to configuration files)
FAU_STG.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
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
FCS_CKM.1	None	
FCS_CKM.2	None	
FCS_COP.1	None	
FDP_ACC.1	None	
FDP_ACF.1	All requests to perform an operation on an object covered by the SFP.	Syscalls *chmod*, *chown*, *setxattr*, *removexattr, link*, *symlink, mknod*, *open*, *rename*, *truncate*, *unlink*, *rmdir*, *mount*, *umount*, *msgctl*, *msgget*, *semget*, *semctl*, *semop*, *semtimedop, shmget*, *shmctl*; details include identity of object
FDP_ETC.1	LSPP mode only: All attempts to export information	Syscalls *open, mount, umount, accept, connect, sendto, sendmsg*

Component	Event	Details (Event Names)
FDP_ETC.2	LSPP mode only: All attempts to export information	Syscalls *open, mount, umount, accept, connect, sendto, sendmsg* as well as specific audit records created by trusted programs (like star) that export data with its labels. Audit messages from the print spooler indicating printing of labeled data.
FDP_ETC.2	LSPP mode only: Overriding of human-readable output marking (Additional)	The TOE will prohibit overriding of human-readable output markings on printed output. Attempts to do so can be audited by the trusted printer spooler
FDP_IFC.2	None	
FDP_IFF.2	LSPP mode only: All decisions on requests for information flow	System calls operating on objects return failure (EACCES) if information flow was denied, other error codes or success indicate that the information flow was permitted.
FDP_ITC.1	LSPP mode only: All attempts to import user data, including any security attributes	Syscalls *open, mount, umount, accept, connect, sendto, sendmsg*
FDP_ITC.2	LSPP mode only: All attempts to import user data, including any security attributes	Syscalls *open, mount, umount, accept, connect, sendto, sendmsg* as well as specific audit records created by trusted programs (like star) that export data with its labels
FDP_RIP.2	None	
Note 1	None	
FDP_UCT.1	None	
FDP_UIT.1	None	
FIA_ATD.1	None	
FIA_SOS.1	Rejection or acceptance by the TSF of any tested secret.	Event “PAM authentication” (from PAM framework); details include origin of attempt (terminal or IP address as applicable)
FIA_UAU.2	All use of the authentication mechanism.	Event “PAM authentication” (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 “PAM authentication” and “PAM bad_ident” (from PAM framework)

Component	Event	Details (Event Names)
FIA_USB.1	Success and failure of binding user security attributes to a subject (e.g. success and failure to create a subject).	Event "PAM session open" (from PAM framework); syscalls *fork, vfork* and *clone* Failure: Events "PAM authentication" and "PAM bad_ident" (from PAM framework, failure status)
FMT_MSA.1	All modifications of the values of security attributes.	Syscalls *chmod*, *chown*, *setxattr*, *msgctl*, *semctl*, *shmctl*; *open* syscall on SELinux interface files */proc/self/attr/current* and */selinux/load*
FMT_MSA.2	None	
FMT_MSA.3	Modifications of the default setting of permissive or restrictive rules. All modifications of the initial value of security attributes.	Syscalls *umask*, *open*; *open* syscall on SELinux interface files */proc/self/attr/exec, /proc/self/attr/fscreate, /selinux/load*
FMT_MTD.1 Audit Trail	All modifications to the values of TSF data.	Syscalls *open*, *rename*, *link*, *unlink*, *truncate* (of audit log files)
FMT_MTD.1 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 User Attributes	All modifications to the values of TSF data. This needs to include the creation and deletion of users.	Audit text messages from "shadow-utils" trusted programs, details include new value of of the TSF data
FMT_MTD.1 Authentication Data	All modifications to the values of TSF data.	Audit text messages from "shadow-utils" trusted programs; attempts to bypass trusted programs detected through audited syscalls *open*, *rename*, *truncate*, *unlink*
FMT_MTD.1	Management of Roles	Audit text messages from *semodule* and *load_policy* utilities; *open* syscall on the SELinux interface file */selinux/load*
FMT_MTD.3 Secure TSF Data	All rejected values of TSF data	Audit text messages from PAM indicating rejection of an attempt to select a weak password
FMT_REV.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	All modifications to the values of TSF data.	System calls *chmod*, *chown*, *setxattr*, *unlink*, *truncate*, *msgctl*, *removexattr , semctl*, *shmctl*
FMT_SMF.1	None (covered by other management functions)	

Component	Event	Details (Event Names)
FMT_SMR.2	Modifications to the group of users that are part of a role including: • Assignment of users to roles • Assignment of privileges to roles • Creation of roles • Deletion of roles • Deletion of privileges from roles	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). Audit messages from the *semanage* tool. Audit text messages from the *semodule* and *load_policy* tools indicating definitions of new custom roles or modifications of custom roles.
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.
FMT_SMR.2	Unsuccessful attempts to use a role due to the given conditions on the roles	Event: audit text messages from *newrole, login, sshd, su* programs
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	Failure of the TSF	Event: audit text messages from programs in "policycoreutils" suite, including *load_policy, restorecon, fixfiles, newrole*; audit text messages from policy aware programs using *libselinux*: *login, sshd, su, crond*
FPT_RCV.1	the fact that a failure or service disconuity occurred	Event: audit text message from *init* indicating switch to single-user run level
FPT_RCV.1	resumption of the regular operation	Event: audit text message from *init* indicating switch to multi-user run level
FPT_RCV.1	type of failure or service discontinuity	Event: audit text message from the program initiating the switch to single-user mode via *libselinux*: *auditd, init, load_policy*
FPT_RCV.4	if possible, the impossibility to return to a secure state after failure of a security function	Event: audit text message from *init* indicating a failure to switch run levels

Component	Event	Details (Event Names)
FPT_RCV.4	if possible, the detection of a failure of a security function	Event: audit text message from *init* indicating switch to single-user run level
FPT_RVM.1	None	
FPT_SEP.1	None	
FPT_STM.1	Changes to the time.	Event: syscalls *settimeofday*, *adjtimex, clock_settime*
FPT_TST.1	Execution of the TSF self tests and the results of the tests	Event: audit text message from the *rbac-self-test* program
FTA_LSA.1	All attempts at selecting a session security attributes	Event: audit text messages from role aware programs via *libselinux*: *login, sshd, su, crond*
FTA_TSE.1	All attempts at establishment of a user session	Event: audit text messages from role aware programs via *libselinux*: *login, sshd, su, crond*
FTP_ITC.1	Set-up of trusted channel	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 user 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 setuid or setgid program.

5.1.1.3 Audit Review (FAU_SAR.1)

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

a) Date and Time of Audit Event

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

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

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

e) The User Session Identifier or Terminal Type

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

Application Note: The TOE is configured to restrict direct access to the audit records to a user in the audit administrator role.

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, RBAC and MAC controls ensure that only users in the audit administrator role 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) Subject sensitivity label

c) Object sensitivity label

d) group identifier (real and effective)

e) event type

f) outcome (success/failure)

g) login from specific remote hostname

h) **login user id**

i) process id

j) Role that enabled access

k) Date and Time of Audit event

l) Object name

m) **Type of access**

n) Any combination of the above items

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) Object identity

c) Event type

d) Subject sensitivity label

e) Object sensitivity label

f) Users belonging to a specified role

g) Access types on a particular object

h) system call number

i) directory or file name.

j) subject identity (process ID)

k) host identity

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/init.d/audit* script with the 'restart' 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.

Application Note: The system does not support distributed audit in the evaluated configuration, therefore the host identity for all audit records on a specific host is always that host. The system supports defining different audit rules on different hosts which is equivalent to filtering by host identity in a non-distributed environment.

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 and MAC 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 audit 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 **stop all processes that attempt to generate an audit record** if the audit trail is full.

Application Note: The TOE stops processes that want to generate an audit entry when the queue used for audit entries in the kernel is full. This queue will be continuously emptied by the audit daemon and the stopped processes will be resumed when there are empty entries in the queue. If the audit trail itself gets full, the audit daemon will not be able to empty the queue, and the audit daemon will execute an audit 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 audit administrator can then back up the audit trail and make space available for the audit trail, then restart the TOE in multiuser mode.

5.1.2 Cryptographic Support (FCS)

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 [SSLv3]** 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 [SSLv3]. 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 repect 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.

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 v2.0 standard SSH - [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 v2.0 standard using the Diffie-Hellman key negotiation protocol.**

Application Note: For details of the key generation / key negotiation process see section 4.5, chapter 5 and chapter 6 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 v2.0 standard. 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.

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. 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 primality tests and the probability of the numbers chosen to be prime.

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.

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 Internet Draft: SSH Transport Layer Protocol [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.

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: Internet Draft: SSH Transport Layer Protocol [SSH-TRANS]**.

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.

Cryptographic operation (RSA) (SSL: 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.

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 [TLS-AES].**

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 Transport Layer Protocol [SSH-TRANS] and the following cipher suite: 3des-cbc as defined in [SSH-TRANS].**

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 (DAC) 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 (SYSV and POSIX message queues, SYSV semaphores, SYSV 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)

FDP_ACC.1.1 The TSF shall enforce the **Role-based Access Control (RBAC) 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 (SYSV and POSIX message queues, SYSV semaphores, SYSV shared memory segments) and all operations among subjects and objects covered by the RBAC policy.**

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

FDP_ACF.1.1 The TSF shall enforce the **Discretionary Access Control (DAC) 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.

5.1.3.4 Role-Based Access Control Functions (FDP_ACF.1) (2)

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

a) **User identity; and**

b) **Authorized roles for the user**

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)

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 TSC.

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

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

a) Devices used 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) **Only data with the same sensitivity label as the sensitivity label of the device can be exported using the device.**

5.1.3.6 Export of Labeled User Data (FDP_ETC.2)

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 which 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) **No additional rules**.

5.1.3.7 Mandatory Access Control Policy (FDP_IFC.1)

FDP_IFC.1.1 The TSF shall enforce the **Mandatory Access Control Policy** on **tasks operating on behalf of a user**, **file system objects, IPC objects, network objects**, **and all operations among subjects and objects covered by the MAC policy.**

5.1.3.8 Mandatory Access Control Functions (FDP_IFF.2)

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 via 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, then 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; then 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; then the flow of information from subject B to subject A is permitted.

FDP_IFF.2.3 The TSF shall enforce the **no additional rules**.

FDP_IFF.2.4 The TSF shall provide the following **no additional rules**.

FDP_IFF.2.5 The TSF shall explicitly authorize an information flow based on the following rules: **trusted subjects with MLS override capabilities can access objects without being restricted by subject and object labels. Trusted objects can be accessed by any subject**.

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 super-set 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.

Application Note: The TOE enforces an additional restriction on write operations. The subject and object labels must be equal to ensure integrity. This "write equal" policy is a stricter variant of the "write up" policy described in this SFR.

5.1.3.9 Import of Unlabeled User Data (FDP_ITC.1)

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) **No additional rules**.

5.1.3.10 Import of Labeled User Data (FDP_ITC.2)

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) **No additional rules**

Application Note: LSPP has an additional item "c)" for FDP_ITC.2.5 which defines sensitivity labels as consisting of hierarchical levels and a set of non-hierarchical categories. This is redundant information and does not make sense in the context of rules regarding import. As it appears to be a cut&paste error in the protection profile, it has been omitted from this ST.

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.

Basic data exchange confidentiality (FDP_UCT.1)

FDP_UCT.1.1 The TSF shall enforce the **Discretionary Access Control Policy, Role-based Access Control Policy, and 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 and ssl connections.

Data exchange integrity (FDP_UIT.1)

FDP_UIT.1.1 The TSF shall enforce the **Discretionary Access Control Policy, Role-based Access Control Policy, and 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 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 and ssl 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) User Clearances

e) **List of Security-relevant Roles; and**

f) no other attributes

Application Note: The “List of Security-relevant Roles” corresponds to the “Security-Relevant Roles” [LSPP] and “List of Authorized Roles” [RBACPP].

Application Note: “Authentication data” includes all data needed for successfully authenticating a user or changing the authentication token. This consists of the user’s password, password age, hashes of previously used passwords, and information about locked or expired accounts.

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) **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) **The current role the user is operating with (from the list of roles the user is allowed to operate with).**

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

a) The sensitivity label associated with a subject shall be within the clearance range of the user;

b) 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.

c) 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.

d) The role associated with a subject shall be one of the authorized roles assigned to 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) Roles can be changed by executing trusted programs for which the SELinux policy defines a role transition, such as the newrole program (provided the authentication is successful).

e) Privileged subjects can change their own security attributes.

Application Note: Privileged executables for which the SELinux policy defines a domain or role transition have a SELinux type whose name ends in "_exec_t", for example *newrole_exec_t*.

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 **users in administrative roles allowing modification of access control attributes 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.**

5.1.5.2 Management of Object Security Attributes (FMT_MSA.1) (2)

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 role allowed to modify sensitivity labels of objects.**

5.1.5.3 Management of Object Security Attributes (FMT_MSA.1) (3)

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

(a) User Role Authorizations

5.1.5.4 Management of Object Security Attributes (FMT_MSA.1) (4)

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

(a) Default Active Role Set

5.1.5.5 Management of Object Security Attributes (FMT_MSA.1) (5)

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

5.1.5.6 Management of Object Security Attributes (FMT_MSA.1) (6)

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.

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 requirement is included as a dependency from the security functional requirements FCS_CKM.1, FCS_CKM.2 and FCS_COP.1. The assessment with respect to this requirement in the evaluatiuon 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.

5.1.5.7 Static Attribute Initialization (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 **Discretionary Access Control Policy**.

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

5.1.5.8 Static Attribute Initialization (FMT_MSA.3) (2)

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 **Mandatory Access Control Policy**.

FMT_MSA.3.2 The TSF shall allow the **users in an administrative role** to specify alternative initial values to override the default values when an object or information is created.

Application Note: The term SFP in FMT_MSA.3.1 in Volume 2 of the Common Criteria is printed in italics but is not as one would expected stated as "[assignment: SFP]". It is assumed that such an assignment was intended by the authors of the CC and has therefore been performed here.

5.1.5.9 Static Attribute Initialization (FMT_MSA.3) (3)

FMT_MSA.3.1 The TSF shall enforce the **RBAC SFP** to provide **administrative user defined** default values for security attributes that are used to enforce the **RBAC SFP**.

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

5.1.5.10 Management of the Audit Trail (FMT_MTD.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.

5.1.5.11 Management of Audited Events (FMT_MTD.1)

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.

5.1.5.12 Management of User Attributes (FMT_MTD.1)

FMT_MTD.1.1 The TSF shall restrict the ability to **initialize and modify** the **user security attributes, other than authentication data**, to **users in a properly authorized administrative role**.

5.1.5.13 Initialization of Authentication Data (FMT_MTD.1)

FMT_MTD.1.1 The TSF shall restrict the ability to **initialize** the **authentication data** to **users in a properly authorized administrative role**.

5.1.5.14 Management of Authentication Data (FMT_MTD.1)

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

a) **users in a properly authorized administrative role; and**

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

5.1.5.15 Management of Roles (FMT_MTD.1)

FMT_MTD.1.1 The TSF shall restrict the ability to **modify and 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

5.1.5.16 Secure TSF Data (FMT_MTD.3)

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

Application Note: The TOE implements a password quality checking mechanism which prevents users from selecting weak passwords.

5.1.5.17 Revocation of User Attributes (FMT_REV.1)

FMT_REV.1.1 The TSF shall restrict the ability to revoke security attributes associated with the **users** within the TSC to **a set of administrative roles**.

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, 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 CAPP Application Note.

5.1.5.18 Revocation of Object Attributes (FMT_REV.1)

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, Role-Based Access Control Policy and Mandatory Access Control policies.**

Application Note: The policies define the rights of object owners and administrative roles authorized to revoke security attributes. The revocation is permitted only if all applicable policies allow the revocation.

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; and

b) **The rules of the Mandatory Access Control policy are enforced on all future operations; and**

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.

Application Note: Like most other UNIX type operating systems the TOE implements delayed revocation as indicated in the CAPP Application Note (cf. section 5.1.5.17 of this document). The next "access to the object" revocation requirement from [RBACPP] and the "all future operations" requirement from [LSPP] are interpreted as referring to the time of the next access check.

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 [CAPP] and was added because a dependency from FMT_MSA.1 and FMT_MTD.1 to this new component has been defined in [CC].

5.1.5.19 Security Management Roles (FMT_SMR.2)

FMT_SMR.2.1 The TSF shall maintain the roles:

a) **Set of RBAC administrative roles;**

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

c) **users authorized by the Mandatory Access Control Policy to modify object security attributes;**

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

e) **users not authorized to modify their own authentication data.**

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

FMT_SMR.2.3 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 (except for the sensitivity label)

(b) The set of RBAC administrative roles can modify security attributes for all objects under the control of TOE (since they automatically inherit the privileges of all Object Owners).

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.

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. The reader should be aware that in the case of System x, System p5 and eServer the abstract machine is the real hardware, while in the case of System z and System p5 the abstract machine is a virtualization of the real hardware by a virtualization layer.

5.1.6.2 Failure with preservation of Secure State (FPT_FLS.1)

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

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)

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)

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.

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. The TOE software can run on many different platforms. All those platforms provide a Memory Management Unit that enforces address space separation between trusted and untrusted subjects; and a multi-state CPU where modification to the address space definition, direct access to peripheral devices, and the CPU configuration itself 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 **MLS labels** when shared between the TSF and another trusted IT product.

FPT_TDC.1.2 The TSF shall use **administrator-defined MLS label mapping rules** when interpreting the TSF data from another trusted IT product.

Application Note: The TOE supports using IPsec security associations (SA) to exchange label information among systems. Note that IPsec encryption and authentication is beyond the scope of this Security Target, it is only considered as a label exchange mechanism.

Application Note: This security functional requirement is not included in [LSPP] and was added because a dependency from FDP_ITC.2 to this new component has been defined in [CC].

5.1.6.9 TSF Self Test (FPT_TST.1)

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)

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)

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**.

Application Note: The system does not permit empty role sets to be specified for a user. Administrators cannot define users without assigning at least one role, and cannot delete a role definition if the system still has users assigned to that role. It is not possible to establish a session with an empty set of roles, therefore this SFR is met implicitly.

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 **communication channel that use the SSH v2.0 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 only 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 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 412 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. Those functions are required as a basis for application level security functions and mechanisms and can be used to build application specific security policies.

The TOE can be operated in two different modes: “LSPP mode” and “CAPP mode”. When in CAPP mode the TOE provides security functions very similar to those that have been evaluated with the previous version of the TOE (Red Hat Enterprise Linux Version 4). In LSPP mode the TOE has activated the SELinux MLS security module, which provides mandatory access control, and is also compliant with the requirements of the role-based access control model requirements as defined in [RBACPP].

The two modes of operation differ by the configuration of the SELinux security module. In addition the TOE requires use of trusted programs appropriate for each of the two different modes. The specific differences are documented in the Evaluated Configuration Guide [ECG].

6.1.2 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
- SYSV semaphores
- SYSV shared memory
- SYSV message queues
- POSIX message queues
- Internet domain sockets
- UNIX domain sockets

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.

Section 2.4.1 of this document lists the file system included in the evaluated configuration.

Ext3 and ISO-9660 represent file systems on a physical medium (disk (ext3), CDROM (ISO-9660)), and the tmpfs file system provides nonpersistent file storage in system RAM.

The other supported file systems do not represent or provide physical data storage file systems but are used as a configuration and monitoring interface to the kernel, provided by the kernel only in a running system.

6.1.3 Non-Kernel TSF Services

The non-kernel TSF services are:

- Identification and Authentication services
- Network application layer services
- Configuration and management commands requiring special privileges

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.

6.1.4 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 that are part of the TSF are built on TCP or UDP. The TCP based application protocols supporting user authentication and running on privileged ports are:

- secure shell (SSH v2.0)
- file transfer services (FTP)

In addition the TOE supports secure socket layer (SSL v3) protocol with the stunnel program, 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.

6.1.5 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:

- The Red Hat Enterprise Linux kernel is running on each computer in the networked system.
- Identification and authentication (I&A) is performed locally by each computer. Each user is required to Login with a valid password and user identifier combination at the local system and also at any remote computer where the user can enter commands to a shell program (using ssh) or use ftp. 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 it defined.

- Discretionary access control (DAC), role-based access control and mandatory access control (when operated in LSPP mode) is performed locally by each of the host computers and is based on user identity, group membership, user roles and the object attraibute on this host. Each process has an identity (the user on whose behalf it is operating), belongs to one or more groups and operates with a role. All named objects have an owning user, an owning group, DAC attributes, 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.
- When operated in LSPP mode, Role-based access control (RBAC) is implemented as part of the SELinux policy. This allows defining a set of roles that can be assigned to users and a set of domains a user in a role can switch to. The TOE includes a policy that defines a hierarchical set of roles with general system administration, security administration and audit configuration assigned to different roles.
- When operated in LSPP mode, the security context assigned to each object and process also contains the sensititvity label of the object or process. Processes get a security context from the user that initiated them. On every access of a process to a protected resource the TOE will evaluate the sensitivity labels of the subject and the object and check if access is allowed according to the rules of the mandatory access control.
- Object reuse is performed locally, without respect to other hosts.
- Interrupt handling is performed locally, without respect to other hosts.
- Privilege is based on the user identity and user role.

6.1.6 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 hardware platforms Red Hat Enterprise Linux is running on support two execution states where kernel mode or supervisor state, software runs with specific privileges to perform operations on the underlying hardware platform and user mode or problem state software runs without those privileges. 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 setuid and setgid programs that can be executed by untrusted users.

6.1.7 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.

6.1.7.1 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 TSF.

- System calls (e.g. open, fork), through which a process requests services from the kernel. They 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 IPC mechanisms, are generally created through use of directly invoked processes (some trusted, some untrusted). A few daemons perform security functions and therefore are part of the TSF interface.
- Network Services, (ssh, ftp, stunnel using ssl, IPsec). 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 on another system. 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 IPsec, ssh, stunnel and ftp are seen as TSF, because they use privileged ports. ssh and ftp require user identification and authentication, and ssh and stunnel 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.

6.1.7.2 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 (direct login at the system console is allowed to avoid a specific 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 the TSF is seen as part of the TSF interface. The files in the administrative database are protected by the different access control mechanisms of the TOE. It is therefore very important to set the access rights and security context to the files of the administrative database such that users in non-administrative roles 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 databases 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. In the TOE administrative tasks are assigned to defined roles allowing a finer grained administrative model based on the role(s) of a user.

6.1.8 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 (in LSPP mode)
- Role-based Access Control (in LSPP mode)
- 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. The following PAM modules are included in the evaluated configuration and implement security functions:

- pam_unix.so (basic password based authentication, configured to use MD5)
- pam_loginuid.so (set permanent audit login user ID, and ensure fail-secure behavior by refusing login in case the audit system is inoperative)
- pam_wheel.so (to restrict the use of the su command to members of the wheel group)
- pam_tally2.so (to limit the number of consecutive unsuccessful authentication attempts)
- pam_nologin.so (to check /etc/nologin)
- pam_securetty.so (to restrict root access to specific terminals)
- pam_passwdqc.so (for additional password checking)
- pam_selinux.so (to set the default security context when establishing a session. When an application opens a session using *pam_selinux.so*, the shell that gets executed will be run in the default security context. The module modifies the security context of the controlling tty to match the one of the user.)
- pam_namespace.so (to establish a private namespace with polyinstantiated directories when establishing a session. Polyinstantiated directories are needed to achieve greater information separation for public use directories such as */tmp* and */var/tmp*, and provide users with writeable home directories as they transition roles, types or sensitivity labels.)

In addition the following module may be used:

- pam_rootok.so (to avoid that an administrative user with the effective user ID of root has to re-enter the password)

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 mechanism for synchronizing this 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 passwords by using the *passwd* command, which is a setuid program with the owning userid of 0. This configuration allows a process running the *passwd* program to read the contents of */etc/shadow* and to modify the */etc/shadow* file for the user's password entry, which would ordinarily be inaccessible to a non-privileged user process (IA1.1). Users are also warned to change their passwords at login time if the password will expire soon, and are prevented from logging in 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 other (not security relevant) information (IA1.3). The encrypted 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 as for example described in the paper of Ken Thomson and Bob Morris „Password Security – A Case History".

The file */etc/shadow* contains the MD5 encrypted 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.

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 databases */etc/shadow* and */etc/security/opasswd* 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 the administrative user to define restrictions on authentication data such as the minimum length of the password, checking the password against entries in a dictionary as well as the maximum life time of a password, the number of unsuccessful login attempts allowed before the account is locked (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 the */var/log/lastlog* file (IA1.6).

In the evaluated configuration the above mentioned parameter need to be set in accordance with the following restrictions:

- 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
- Maximum number of attempts to change the password: 3
- Password history length: 7
  (IA1.7)

When operated in LSPP mode the login mechanism assigns a default sensitivity label for the session. This sensitivity label must be dominated by the clearance of the user (IA1.8). In the case of network login via *ssh*, the label of the network connection must match the sensitivity label of the session (IA1.9).

When operated in LSPP mode, the login mechanism assigns a default role for the session from the list of roles assigned to the user by using the *newrole –r* command (IA1.10).

This function contributes to satisfy the security requirements FIA_ATD.1, FIA_SOS.1, FMT_MTD.1 „User Attributes", 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
- Get user security characteristics (e.g., user and groups)
- Check if the sensitivity label specified by the user for the session is within the range of the sensitivity labels allowed for this user (in LSPP mode only)
- Check if the role specified for the session is within the set of roles assigned to the user (in LSPP mode only)

The common I&A mechanism identifies the user based on the supplied user name, gets that user's security attributes, and performs authentication against the user's password.

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 and Role 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, filesystem 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 setuid 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 setuid or setgid bit set (IA4.6).

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

A user can change his current active role using the *newrole -r* command (IA.4.7). The command requires the user to authenticate himself and allows to change his role to one of the roles assigned to him after successful authentication (IA.4.8).

A user can change his current active clearance using the *newrole –l* command when using non-network terminals (such as a serial console) or for starting noninteractive processes that do not interact with the terminal(IA.4.9). The command requires the user to authenticate himself and allows to change his clearance to a new level and combination of categories from the set of those assigned to him after successful authentication (IA.4.10)

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). 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 (IA.5.2).

When operating in LSPP mode, the role of the user is either the one specified by the user (provided it is within the set of roles assigned to the user) or the user's default role (IA.5.3).

When operating in LSPP mode, the sensitivity label of the user's session is either the one specified by the user (provided it is within the range allowed for the user) or the user's default sensitivity label defined in his user profile (IA.5.4).

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

6.2.2.6 TOE access (IA.6)

When initiating an interactive user session via *login, ftp,* or *sshd*, or running tasks on a user's behalf via *crond*, the system restricts the active role set for the user to the set of authorized roles for that user (IA.6.1). The system enforces that the set of authorized roles for the user is never empty (IA.6.2).

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

6.2.3 Audit (AU)

The Lightweight Audit Framework (LAF) is designed to be a CAPP compliant audit system for Linux. LAF is built on top of systrace which is a system call security policy enforcement engine first developed for BSD but ported to Linux. 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)

The system administrator 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/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/init.d/audit 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 (Management of audited events).

6.2.3.2 Audit Processing (AU.2)

Auditing is performed on a per process basis. A process can enable or disabling auditing for itself by attaching itself or detaching itself to the audit subsystem provided it is running with root 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. An internal queuing mechanism is used for this purpose. When the queue does not have sufficient space to hold an audit record 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 (AU2.8). 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 discretionary access control function 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 and FMT_MTD.1 (Management of the audit trail).

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 timestamp and audit ID to permit easy correlation. When operating in LSPP mode, audit records also contain the role the user currently operates with (AU3.2) and the sensitivity labels of the subject and object (AU.3.3).

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 (AU.3.4).

The TOE maintains a "login ID" which is set when the user performs his initial login at a terminal or via a network connection (AU.3.5). 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 (AU.3.6). This allows tracing 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:

less reads the ASCII audit data (AU.4.1).

ausearch allows selective extraction of records from the audit trail using defined selection criteria (AU.4.2)

The audit records are listed in chronological order by default. The *sort* utility can be used together with *ausearch* to use a different sorting order (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.

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, a subject with a filesystem user ID of 0 is exempt from all restrictions of the discretionary access control and can perform any action desired (DA1.1).

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 root (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).

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 all supported file systems (see section 2.4.1). 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. In addition, file systems do not necessarily support individually configured ownership and rights for files and directories, the permissions may be predefined based on global per-filesystem properties or implicit object properties.)

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)

  Each process has an inheritable "umask" attribute which is used to determine the default access permissions for new objects. It is a bitmask of the user/group/other read/write/execute bits, and specifies the access bits to be removed from new objects. For example, setting the umask to "002" ensures that new objects will be writable by the owner and group, but not by others. (DA2.3)

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

DISCRETIONARY ACCESS CHECK ALGORITHM
A process may request read, write, or execute/search access to a file system object protected by an ACL. The discretionary access check algorithm determines whether access to the object will be granted according to the DAC policy. In LSPP mode, Role-based access control and the sensitivity label based mandatory access control can impose additional restrictions denying access even when the DAC algorithm would allow access.

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) , FIA_USB.1 and FDP_ACF.1(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), FMT_MSA.1(1), FMT_SMF.1, FMT_MSA.3(1) , FIA_USB.1 and FPT_SEP.1.

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), FIA_USB.1 and FPT_SEP.1.

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

6.2.4.4.1 DAC: SYSV 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: POSIX and SYSV 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 an effective user ID of 0 (DA4.16).

6.2.4.4.3 DAC: SYSV 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 an effective user ID of 0 (DA4.25).

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

6.2.5 Role-Based Access Control (RA) (LSPP mode only)

The TOE allows defining roles in the SELinux policy by assigning the domain types to the role to which a user in that role may transition. Each subject has a single active role at all times (RA.1.1). The following roles are defined in the TOE policy in LSPP mode:

Administrative roles:

- system: The operating system supports multiple roles for noninteractive system processes such as daemons. All non-interactive roles are considered to be subdivisions of a conceptual “system” role. The additional restrictions enforced on system services are beyond the scope of this Security Target. The definition of system roles allows separating those from users (RA.1.2).
- sysadm: This is a role defined for general system administration tasks (RA.1.3), including setting or modifying security contexts, and changing the sensitivity label of a subject or object (RA.1.4).
- auditadm: This is a role for the management of the audit configuration and evaluation of the audit records (RA.1.5).

Non-administrative roles:

- Staff: This is a role for users that are allowed use the *newrole* command to transition to administrative roles (RA.1.6).
- User: This is a generic role for all users (as opposed to system processes) (RA.1.7).

Each user has a set of permitted roles and a default role (both defined by the administrator) and may select an active role using the *newrole* command from the set of permitted roles. Rules in the policy also define which transitions between roles are allowed. Role transition requests succeed only if the new role is in the set of permitted roles for the current user, and if the policy allows a transition from the current role to the new role (RA.1.8).

Administrators can define additional roles using SELinux loadable policy modules defined using the *checkmodule*, *semodule_package*, and *semodule* utilities as documented in the Evaluated Configuration Guide (RA.1.9). A role definition consists of a set of permitted role transitions to or from that role, and a set of SELinux domains which correspond to rights associated with the role. Additional roles may be administrative roles with permission to use domains that have specific privileges, including DAC and MAC override capabilities (RA.1.10). The policy tools ensure that role definitions may only be removed from the system if the rule is not included in the permitted rule set for any user.

RBAC access checks are performed whenever a subject accesses an object, with the permission based on the subject’s domain, the object’s type, and the operation attempted. The RBAC policy covers all objects covered by the DAC policy. SELinux “allow” rules define the specific access rights to object types for the domains that are associated with the role. Any access attempt from a domain to an object type that is not explicitly permitted by a SELinux “allow” rule is rejected.

Role-based access checks can veto actions that would normally be permitted by DAC or MAC rules, but can never permit something that would be denied according to DAC or MAC rules (RA.1.11). Access is permitted only if all applicable policies (DAC, RBAC, and MAC in LSPP mode) agree that the access is permitted (RA.1.12).

Whenever an operation would result in an illegal SELinux context for a subject or object, for example an invalid combination of role and SELinux user class, the operation will fail and leave the subject and object properties unchanged (for modifying operations), or refuse creation (for creating operations). This ensures that subjects always have exactly one active role.

This function contributes to satisfy the security requirements FMT_SMR.2, FDP_ACC.1(2), FDP_ACF.1(2), FMT_MSA.1(3,4,5,6), FIA_USB.1, and FMT_MSA.3(3).

6.2.6 Mandatory Access Control (MA) (LSPP mode only)

6.2.6.1 Information Flow Control (MA.1) (LSPP mode only)

When operated in LSPP mode the TOE supports mandatory access control using sensitivity labels automatically attached to processes and objects. This policy is enforced by the SELinux security module and the TOE specific SELinux policy.

Sensitivity labels consist of a hierarchical part (the level) and a nonhierarchical set of categories.

The SELinux security module attaches a “sensitivity label” as part of the security context to the following objects in the kernel (MA.1.1):

- Inodes
- Files
- Directories
- Block devices
- Character devices
- Sockets
- The following list of IPC objects:
    - SYSV Message queues and messages
    - SYSV shared memory
    - SYSV semaphores
    - POSIX message queues and messages
- The following network objects:
    - Ports
    - Network Interfaces
    - Nodes (IP address or netmask)
- Key objects in the kernel key store

Processes are subjects with associated security contexts. When sending signals using the *kill* system call, the process security contexts are used to decide if the sending (active) process has permission to do so.

In addition a task as a subject in the kernel also has a security context attached (MA.1.2). Each process has an effective or “low” sensitivity label (consisting of a hierarchical level and zero or more categories), and a separate “process clearance” or “high” sensitivity label which must dominate the effective label. The effective level is used for all access checks except for processes with the “mlsreadtoclr” override attribute. Access control is performed based on the sensitivity labels of the task and the object. When the tasks attempts a write access to the object, access is granted by the mandatory access control policy only when the effective sensitivity label of the task is equal to the sensitivity label of the object (MA.1.3). (This is a stricter variant of the “write up” policy required by FDP_IFF.2.) Read access is granted by the mandatory access control policy only when the effective sensitivity label of the task dominates the sensitivity label of the object (MA.1.4). In addition the access needs to be granted by the role the user associated with the task is operating with as well as by the discretionary access control algorithm. When in LSPP mode all three access control policies implemented by the TOE must allow access before the operation attempting to access the object is allowed to proceed.

Attaching the security context to those objects, evaluating the security context in case of access attempts and managing the security context of subjects and objects is performed by functions that SELinux provides for the kernel hooks defined in the LSM framework. The functions at those hooks ensure that all subjects and objects get a security context (including a sensitivity label) when they are created in accordance with the rules of the mandatory access control policy. The functions of SELinux also ensure that the sensitivity labels are evaluated whenever a task performs a function that accesses one of the objects listed above. This is implemented as SELinux constraints which veto any access that violates the MAC policy. These constraints cannot be removed by administrator defined local policy modifications when those are performed in accordance with the Evaluated Configuration Guide [ECG].

The MAC policy constraints define specific override capabilities for trusted subjects and objects as follows (MA.1.5):

Attribute *Used in "typeattribute" statements (*.if files) and "mlsconstrain" rules ("policy/mls" file)*	**Interface** *Refpolicy macros used in *.te files to give the right to specific domains*	**Description** *From the "policy/modules/kernel/mls.if" refpolicy file.* *"MLS trusted" means the operation is permitted by the MLS access check, but may still be denied by the DAC or RBAC policy.* *"proc" or "process" read/write refers to pseudofiles in the proc file system for processes other than the current process.* *"files" includes all filesystem objects other than procfs pseudofiles.* *"higher" means "not dominated by the subject level" and includes incomparable labels.* *"lower" means "not dominating the subject level" and includes incomparable labels.* *"process clearance" is the high level for the process (as opposed to the "low" level which is the effective level for all access checks except the "readtoclr" operation).*
mlsfileread	*mls_file_read_up*	Make specified domain MLS trusted for reading from files at higher levels.
mlsfilewrite	*mls_file_write_down*	Make specified domain MLS trusted for writing to files at lower levels.
mlsfileupgrade	*mls_file_upgrade*	Make specified domain MLS trusted for raising the level of files.
mlsfiledowngrade	*mls_file_downgrade*	Make specified domain MLS trusted for lowering the level of files.
mlsnetread	*mls_socket_read_all_levels*	Make specified domain MLS trusted for reading from sockets at any level.
mlsnetreadtoclr	*mls_socket_read_to_clearance*	Make specified domain MLS trusted for reading from sockets at any level that is dominated by the process clearance.
mlsnetwrite	*mls_socket_write_all_levels*	Make specified domain MLS trusted for writing to sockets at any level.
mlsnetrecvall	*mls_net_receive_all_levels*	Make specified domain MLS trusted for receiving network data from network interfaces or hosts at any level.
mlsipcread	*mls_sysvipc_read_all_levels*	Make specified domain MLS trusted for reading from System V IPC objects at any level.
mlsipcwrite	*mls_sysvipc_write_all_levels*	Make specified domain MLS trusted for writing to System V IPC objects at any level.
privrangetrans	*mls_rangetrans_source*	Allow the specified domain to do a MLS range transition that changes the current level.
mlsrangetrans	*mls_rangetrans_target*	Make specified domain a target domain for MLS range transitions that change the current level.
mlsprocread	*mls_process_read_up*	Make specified domain MLS trusted for reading from processes at higher levels.
mlsprocwrite	*mls_process_write_down*	Make specified domain MLS trusted for writing to processes at lower levels.
mlsprocsetsl	*mls_process_set_level*	Make specified domain MLS trusted for setting the level of processes it executes.
mlstrustedobject	*mls_trusted_object*	Make specified object MLS trusted.

(The shipped MLS policy includes definitions for X11 objects which are not relevant for the evaluated configuration.)

Trusted subjects are programs launched by administrators and trusted programs running with elevated privileges, as indicated by domains with MLS type attributes such as “mlsfileread”.

Trusted objects (with object type attribute “mlstrustedobject”) are pseudofiles that do not actually store data and may therefore override MLS access restrictions, for example the */dev/null* and */dev/zero* devices which need to be accessible to processes at all sensitivity levels, and */dev/tty* which is an alias for the current terminal device.

Whenever an operation would result in an illegal SELinux context for a subject or object, for example an invalid MLS sensitivity label, the operation will fail and leave the subject and object properties unchanged (for modifying operations), or refuse creation (for creating operations). This ensures that subjects and objects always have a valid sensitivity label.

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

6.2.6.2 Import/Export of labeled data (MA.2) (LSPP mode only)

The system supports import and export of unlabeled data from/to single level devices. Changes in device level must be performed manually by the administrator and are auditable. (MA.2.1)

The *star* tool permits import and export of labeled filesystem data when used by administrators by creating archives that preserve label information (MA.2.2).

The print spooler converts input into bitmaps and adds human readable labels to the bitmaps based on the input data label. The final bitmap is encapsulated into either PCL 4 or PostScript level 1 (depending on the print queue configuration) and sent to the printer via a parallel or USB interface. The printer must support the configured printer language. (MA.2.3).

The TOE IPsec and CIPSO implementations allow assigning labels to network objects and enforcing the mandatory access control policy based on those labels. (MA.2.4)

The IPsec implementation can be used for encrypted and authenticated network communication which is beyond the scope of this Security Target. IPsec is only supported for the purpose of labeled networking, and only in transport mode. Tunnel mode is not supported.

Trusted programs are used to administer the IPsec policy: *setkey, racoonctl* and *racoon*. *setkey* can be used to edit the security policy database and specify when which security association is to be used in the communication between two systems (e. g. for which ports). *racoon* is a daemon that can set up security associations automatically. The *racoon* daemon can be administered using the *racoonctl* trusted program.

The CIPSO policy is configured using the *netlabelctl* trusted program.

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

6.2.7 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.

Note that ext3 and ISO 9660 are the only supported disk based filesystems. All other filesystems do not have persistent backing storage and therefore object reuse of disk space is not an issue for them.

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.

The procfs, sysfs, selinuxfs, binfmt_misc, and rootfs file systems only provide a very limited view of specific in-kernel data structures and cannot be used for arbitrary data storage. Object reuse is handled inside the kernel when allocating and deallocating data structures.

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.

In addition to specific utilities mentioned in this section, administrators can use the *rnano* editor to modify configuration files and scripts when the system does not supply a specific trusted program designed to do so.

6.2.8.1 Roles (SM.1)

The TOE maintains a hierarchical set of roles with some administrative roles and two user roles as defined in section 6.2.5 of this document (SM1.1).

In the evaluated configuration, the set of administrative users consists of those with permission to use the *newrole* utility to switch to an administrative role. Every administrative user has a unique personal userid to log into the system. This helps to provide accountability and to prevent misuse of privileges. The userid "root" cannot be used for direct login except for login from the system console.

Using the SELinux role-based access control model each user may be authorized for a set of roles, and each role is authorized for a set of type enforcement (TE) domains. A role dominance relationship can optionally be specified in the configuration to define a hierarchy among roles. The assignment of permissions is primarily deferred to the TE configuration. This approach combines the ease of management provided by the RBAC model with the fine-grained protections provided by the TE model.

The SELinux RBAC model maintains a role attribute in the security context of each process. For objects, the role attribute is typically set to a generic object_r role and is unused.

SELinux maintains a user identity attribute in the security context that is independent of the Linux user identity attributes. The policy configuration limits the ability to change the SELinux user identity attribute to certain TE domains. These domains are associated with certain programs, such as login, crond, and sshd, that have been modified to call functions from *libselinux* to set the SELinux user identity appropriately. Hence, user login sessions and cron jobs are initially associated with the appropriate SELinux user identity, but subsequent changes in the Linux uid may not be reflected in the SELinux user identity. In some cases, this is desirable in order to provide user accountability or to prevent security violations.

In LSPP mode, even the administrative roles are subject to MAC checks. The exception is the special Unconfined role which can be used to selectively circumvent MAC restrictions, this role is by default not made available to administrators.

6.2.8.1.1 Administrative Users

Users that are allowed to use the *newrole* command to switch to an administrative role can perform administrative actions. Users that don't have the privilege to use *newrole* to switch to an administrative role can not perform administrative actions. Users that are not member of the trusted group can also not login as root even if they know the root password (SM1.2).

6.2.8.1.2 Normal Users

Normal users can not perform actions that require administrative privileges. They can only execute those setuid root programs they have access to (SM1.3). 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. Note that the use of passwd to change the own password may be prohibited by the user's role.

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

6.2.8.2 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).

In LSPP mode specifically authorized users can modify the sensitivity labels of objects using the chcon command.

This function contributes to satisfy the security requirements FMT_MSA.1, FMT_MSA.3, FMT_SMF.1 and FMT_REV.1 „Object Attributes".

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

6.2.8.3.1 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)

Additional user attributes such as the set of permitted roles and security labels (level range and permitted categories) are stored in the */etc/selinux/mls/seusers* and */etc/selinux/mls/users/** files.

6.2.8.3.2 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 including the user roles (mainly */etc/passwd, /etc/selinux/mls/users/*,* and *etc/selinux/mls/seusers*) (SM3.3).

6.2.8.3.3 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
  (SM3.4)

All users except those only have the "user" role are also allowed to change their own password using the *passwd* command. The password restrictions defined by the administrative user apply (SM3.5).

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 „User Attributes", FMT_MTD.1 „Authentication Data", FMT_SMF.1 and FMT_REV.1 „User Attributes".

6.2.8.4 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/init.d/audit* 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 *auditd* command and the */etc/init.d/audit* 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/init.d/audit* 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).

This function contributes to satisfy the security requirements FAU_GEN.1 and FAU_SEL.1 as well as FMT_MTD.1 (Management of the audit trail) and FMT_MTD.1 (Management of audited events)

6.2.8.5 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 a system administrator in the *sysadm* role 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 a system administrator in the *sysadm* role 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

6.2.9 Secure Communication (SC)

The TOE provides the ability to protect communication by cryptographic mechanism against disclosure and undetected unauthorized modification. The TOE supports protocols (SSH v2.0 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 cryptography used in this product has not been FIPS 140 certified. This Security Target claims compliance with the external standard for the cipher suites explained by the SFRs of FCS_COP.1 including all its iterations for the definition of the encryption algorithm. There are many ways of determining compliance with a standard. The vendor has chosen to make a developer claim of compliance supported with 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.

The protocols SSH v2.0 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 users 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 access control mechanisms.

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). Those protocols are:

- the Secure Shell Transport Layer Protocol Version 2 [SSH-TRANS] and the Secure Shell Authentication Protocol [SSH-AUTH]
- the Secure Socket Layer Protocol Version 3 [SSLv3]

The SSH and SSL protocols are able to establish a secure channel between a client and a server process (SC1.1). The TOE supports both the client as well as the server processes for both of those protocols and therefore is able to initiate a connection as well as act as the receiver part. Both protocols provide the ability to “tunnel” an otherwise unprotected single port TCP based protocol.

6.2.9.1.1 The Secure Shell Protocol

The TOE provides the Secure Shell Protocol Version 2 (SSH v2.0) to allow users from a remote host to establish a secure connection and perform a logon to the TOE.

The following table documents implementation details concerning the OpenSSH 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
[SSH-TRANS] 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] 6.2	Compression	OpenSSH supports the OPTIONAL “zlib” compression method.
[SSH-TRANS] 6.3	Encryption	The ciphers supported in the evaluated configuration are detailed below.

[SSH-AUTH] 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] 8.	Password Authentication Method: “password”	This SHOULD authentication method is supported by OpenSSH and is the only authentication method used in the evaluated configuration.
[SSH-AUTH] 8.	Password change request and setting new password	The OpenSSH implementation supports the optional password change mechanism in the evaluated configuration.
[SSH-AUTH] 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 v2.0 protocol:

1. Establishing a secure communication channel using the following cryptographic functions provided by the SSH v2.0 protocol:
   - Encryption using three key Triple DES in CBC mode (3des-cbc as defined in section 4.3 of [SSH-TRANS]) (SC1.2)
   - Diffie-Hellman key exchange (diffie-hellman-group1-sha1 as defined in section 6.1 of [SSH-TRANS]) (SC1.3)
   - The keyed hash function hmac-sha1 for integrity protection as defined in section 4.4 of [SSH-TRANS] (which refers to [HMAC] for the exact definition of the algorithm) (SC1.4).

     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 (Password Authentication Method as defined in chapter 5 of [SSH-AUTH]) (SC1.5).

   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 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 (SC1.6).

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 to tunnel other TCP based protocols (that satisfy the restrictions defined in the Evaluated Configuration Guide) securely between a client and a server system.

The following table documents 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

		below.
[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 below.

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 [SSLv3] or RFC 3268 [TLS-AES]:

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 (SC1.7)

Other cypher suites as defined in [SSLv3] or [TLS-AES] 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 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.

This function contributes to satisfy the security requirements FCS_CKM.1 (1-3), FCS_CKM.2 (1-4), FCS_COP.1 (1-3), FDP_UCT.1, FDP_UIT.1, FMT_MSA.2 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 and process isolation mechanisms. In the evaluated configuration, type enforcement rules ensure that files that are part of the TSF database as well as files and directories containing internal TSF data (e.g. batch job queues) are also protected from unauthorized modification and reading. The type enforcement rules allow access to those files only to roles authorized for access to those types (TP1.2). In addition DAC access control can be defined for additional protection.

The TSF including 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 (TP1.3).

Resources managed by the kernel software can only be manipulated while running in kernel mode (TP1.4).

Processes run in user mode and can call functions of the kernel only as the result of an exception or interrupt (TP1.5). 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 (TP1.6). Other trusted process interfaces are started during system initialization and use well defined protocol or file system mechanisms to receive requests (TP1.7).

Some system calls or parameter of system calls are reserved are reserved for trusted processes. When called the kernel checks that the calling process runs with an effective userid of 0 (TP1.8).

The TOE includes the SELinux framework, and gets control via the LSM hooks in the kernel. With these hooks SELinux gets control each time a named object or a process (as a subject) is created and each time a subject wants to access a named object. SELinux attaches security attributes to each process and each named object and uses rules defined in a policy file to define the default initial values of those attributes as well as to evaluate if a subject may be granted access to an object. Those rules are evaluated in addition to the discretionary access control rules enforced by other kernel subsystems (e. g. a subsystem implementing a file system).

In the evaluated configuration there are two different set of SELinux policy rules for the two modes of operation. In CAPP mode the policy rules enforce the DAC policy with some additional restrictions beyond the scope of this ST. In LSPP mode there are additional rules that define the sensitivity label based mandatory access control policy and the role-based access control policy.

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.

A specific kind of kernel module is SELinux which is implemented as a kernel module of the LSM (Linux Security Module) framework. This framework provides a large number of hooks in the Linux kernel where an LSM can intercept kernel functions and perform additional checks or define and manage the security context of a task or an object. Contrary to other loadable security modules, SELinux is already compiled into the kernel. The SELinux part of the TOE uses those hooks to implement the role-based and mandatory access control policies definable using a policy file, which is separately compiled and then loaded at system boot time.

The TOE ensures that every process is running in a “security domain” and every protected resource has a “type” associated with it. Policy rules define the actions a domain can perform on a type. User roles are defined by the domains a user can use. When in LSPP mode the TOE has a policy file which also includes the rules that define classical mandatory access control for controlling information flow as well as different roles in line with the requirements of a general role-based access control model.

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 root privileges.

A trusted process is distinguished from other user processes by the ability to affect 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 all the kernel interfaces available for their use, 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 root 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 mechanism, if the process has the required access rights for the attempted type of access.

Any program executed with root 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 root privileges (TP4.2).

Trusted processes are not part of the kernel and (except for those processes that perform system initialization and identification and authentication) not part of the TSF itself.

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 6-4 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 6-4 . Administrative Databases. This table lists other administrative files used to configure the TSF.

File Name	Purpose
/etc/aide.conf	Configuration file for AIDE utility
/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/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 separated 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	This file contains the rules for the Netlabel subsystem Each line contains just the arguments to the netlabel command
/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	Configuration file for the racoon IKE daemon, including security association definitions and security policy
/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	Configuration file for the RBAC self test utility
/etc/selinux/config	Defines active policy
/etc/selinux/mls/contexts/	Default file contexts
/etc/selinux/mls/modules/	Folder for the policy modules in LSPP mode
/etc/selinux/mls/policy/	Folder for the policy in LSPP mode
/etc/selinux/mls/setrans.conf	MLS label translations from internal to admin defined names
/etc/selinux/mls/seusers	User sensitivity labels (clearances)
/etc/selinux/semanage.conf	Configuration for the semanage tool
/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 (Server only)
/etc/vsftpd/vsftpd.conf	Contains configuration parameter for the vsftp server (Server only)
/etc/xinetd.conf	Main configuration file for xinetd
/etc/xinetd.d/*	Subsidiary configuration files for xinetd, read from xinetd.conf
/var/lib/aide/aide.db.gz	Program checksum information database for aide utility
/var/lib/aide/aide.db.new.gz	Program checksum information database for aide utility
/var/log/lastlog	Stores time and date of last successful login attempt for each user.
/var/log/tallylog	Stores number of unsuccessful login attempts 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 and FMT_MTD.1 (Management of User Attributes;Authentication Data; and Roles), FMT_MTD.3, FMT_SMR.2, and 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 the system administrator that allows him to test the correct functions of the protection features of the underlying abstract machine. This tool performs tests on

- the main memory (to check for failures in the memory hardware) (TP7.1)
- the processor (to check the functions of the memory management unit and the separation between user and kernel mode) (TP7.2)
- 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 Mechanisms (TP.8)

The TOE provides the *rbac-self-test* tool for the system administrator that allows him to run a system self test to demonstrate correct operation of the TSF. This tool performs tests on

- the integrity of TSF data, including the SELinux policy (TP.8.1)
- the integrity of stored TSF executable code (TP.8.2)
- In LSPP mode: correct operation of the MAC mechanism (TP.8.3)

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 (TP8.4).

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

6.2.10.9 Secure failure state (TP.9)

The system provides a single user maintenance mode. If an operation on the SELinux policy fails, such as any *libselinux* operation that requires access to the policy or role information, the operation is aborted (TP.9.1). The system can be configured to automatically enter single user mode when the self test utility detects a security failure (TP.9.2).

In single user mode, all interactive user sessions are terminated and all system daemons that can run tasks on a user's behalf (*crond*) are unavailable (TP.9.3).

An authorized system administrator can use the system console to interact with the system and re-enter normal multiuser mode (TP.9.4).

This function contributes to satisfy the security requirements FPT_FLS.1, FPT_RCV.1, FPT_RCV.4, and FPT_RVM.1.

6.3 Supporting functions not part of the TSF

6.3.1 User Processes

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 augmented by ALC_FLR.3 are met by Red Hat Enterprise Linux.

Table 6-5: Mapping Assurance Requirements to Documentation

Assurance Component	Documentation describing how the requirements are met
ACM_AUT.1	Configuration management procedures within Red Hat are highly automated using the Red Hat build system.
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 and test results are maintained under configuration management.
ADO_DEL.2	Red Hat Enterprise Linux is delivered on CD / DVD in shrink-wrapped package to the customer. Red Hat verifies the integrity of the production CDs / DVDs by checking a production sample. Other packages that contain fixes must be downloaded from the Red Hat web site. Since those packages are digitally signed the user is able and has to verify the integrity and authenticity of those packages. Every user can verify the integrity of the packages of the

Assurance Component	Documentation describing how the requirements are met
	distribution by verifying their digital signature.
ADO_IGS.1	Guidance for installation and system configuration is provided in the set of guidance documentation provided with the product.
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 spreadsheet 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_LLD.1	A document desribing the low-level design of the TSF is provided. This document contains detailed descriptions of the modules that make up the TSF, their interfaces and interrelationships. Also the commonly used data structures are explained. In addition call graphs show the flow and interdependencies between the modules.
ADV_IMP.1	As an Open Source product the TOE is delivered with its full source code.
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.
ADV_SPM.1	An informal security policy model of the security functions of the TOE is provided as a separate document.
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. Those are augmented by documentation specific for the evaluated configuration.
AGD_USR.1	The Step-by-Step Guide, the Evaluated Configuration Guide and the Reference Guide contain information for users of the TOE. Those are augmented by documentation specific for the evaluated configuration.
ALC_DVS.1	The Red Hat security procedures are defined and described in Red Hat internal documents provided to the evaluation facility.
ALC_FLR.3	The defect handling procedure Red Hat has in place for the development of Red Hat Enterprise Linux requires to describe the defect with its effects, security implications, fixes and required verification steps.
ALC_LCD.1	The Open Source life cycle model with the activities performed by Red Hat as the distributor is provided, showing how the critical aspects of software development and maintenance are covered within this model.
ALC_TAT.1	The tools used for development and testing are defined in a separate document that also explains, how the tools are used within the development and maintenance process.
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 the techniques he used for the vulnerability analysis, the input he used 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 functions is SOF-medium.

7 Protection Profile Claims

7.1 PP Reference

This Security Target claims conformance with the „Controlled Access Protection Profile" [CAPP]. This Protection Profile is listed on the TPEP web site of NSA as a “Certified Protection Profile”, and was also used as the basis of the evaluation of AIX 5.2, Sun Solaris 8 and HP-UX 11.11.

This Security Target also claims conformance with the „Labeled Security Protection Profile" [LSPP] when the TOE is operated in LSPP mode. This Protection Profile is also listed on the TPEP web site of NSA as a “Certified Protection Profile”.

This Security Target also claims conformance with the “Role-Based Access Control Protection Profile” [RBACPP] when the TOE is operated in LSPP mode. This Protection Profile is *not* listed as a certified protection profile.

7.2 PP Tailoring

One additional security functional requirement (FMT_SMF.1) has been added to those defined in the protection profiles. 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.

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 [CAPP], [LSPP], and [RBACPP].

All other security functional requirements in this ST are inherited from the protection profiles and the operations allowed / required by the PP are performed and indicated in bold letters. Two security functional components (FIA_UAU.1 and FIA_UID.1) have been replaced by hierarchical higher ones (FIA_UAU.2 and FIA_UID.2). In both cases the only difference is the fact that no interaction with the TOE is allowed without proper user identification and authentication. This does not modify any of the rationale provided in the PP. The same assessment applies to the use of the hierarchical SFR FMT_SMR.2 as a replacement for FMT_SMR.1.

Security Functional Requirements have been refined where required by the Protection Profiles.

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.

Threats have been added (the Protection Profiles only defines policies). One assumption on the TOE environment (A.NET_COMP) has been added to reflect the distributed nature of the TOE.

One additional security objectives for the TOE (O.COMPROT) has been defined to reflect the ability of the TOE to connect with trusted IT products via trusted channels. Objectives for the TOE environment have been added to this ST in addition to the ones contained in the protection profiles to allow a more distinguished description of the TOE environment - this does not impact the conformance of this ST to the PP.

The following security objectives for the TOE environment have been added:

OE.ADMIN	OE.INFO_PROTECT
OE.MAINTENANCE	OE.RECOVER
OE.SOFTWARE_IN	OE.SERIAL_LOGIN
OE.PROTECT	

Those 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.

The assurance requirements of the Protection Profile are those defined in the Evaluation Assurance Level EAL3 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 [CAPP] and [LSPP] 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.

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.

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.ACCESS, P.NEED_TO_KNOW
O.MANDATORY_ACCESS	P.CLASSIFICATION
O.RESIDUAL_INFO	P.NEED_TO_KNOW, T.ACCESS
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.ASSET, A.ACCESS, A.OWNER, A.CLEARANCE, A.SENSITIVITY
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

Table 8-3: Mapping threats to objectives

Threat	Objective
T.UAUSER	O.AUTHORIZATION, O.MANAGE
T.ACCESS	O.DISCRETIONARY_ACCESS, O.RESIDUAL_INFO
T.COMPROT	O.COMPROT
T.OPERATE	O.MANAGE
T.ROLEDEV	O.DUTY, O.ROLE, O.HIERARCHICAL
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.ASSET	OE.INFO_PROTECT
A.LOCATE	OE.PHYSICAL
A.PROTECT	OE.INFO_PROTECT, OE.PHYSICAL
A.ACCESS	OE.INFO_PROTECT
A.MANAGE	OE.ADMIN, OE.INSTALL, OE.RECOVER
A.OWNER	OE.INFO_PROTECT
A.NO_EVIL_ADMIN	OE.ADMIN, OE.INSTALL
A.COOP	OE.CREDEN
A.UTRAIN	OE.INFO_PROTECT
A.UTRUST	OE.INFO_PROTECT
A.CLEARANCE	OE.INFO_PROTECT
A.SENSITIVITY	OE.INFO_PROTECT
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.ACCESS	O.ROLE
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.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.ACCESS: 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 IT asset compromise due to improper administration and operation of the TOE is removed by O.MANAGE providing functions and facilities necessary to support the administrative users responsible for the management of TOE security.

T.ROLEDEV: The threat of developing and assigning user roles in a way that undermines security is removed by O.DUTY which provides the ‘separation of duties’ capability, O.HIERARCHICAL which supports defining roles in terms of other roles, and O.ROLE which limits access to and operations on resources and objects to members of authorized roles that permit those operations.

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.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.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.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 systems and OE.PHYSICAL requiring physical protection.

A.UTRAIN: The assumption on trained users is covered by OE.INFO_PROTECT which requires that users are trained to protect the data belonging to them.

A.UTRUST: The assumption on user to be trusted to protect data is covered by OE.INFO_PROTECT which requires that users are trusted to use the protection mechanisms of the TOE adequately to protect their data.

A.ASSET: The assumption on the value of the stored assets meriting moderately intrusive attacks is covered by OE.INFO_PROTECT which requires that protection mechanisms are configured properly.

A.ACCESS: The assumption that roles accurately reflect the user's job function, responsibilities, qualifications, an/or competencies within the enterprise is covered by OE.INFO_PROTECT which requires that administrators are trained to perform configuration tasks properly.

A.OWNER: The limited right of users to create and manage new data object is covered by OE.INFO_PROTECT which requires that users are trained to perform these tasks properly and not to pass on information to somebody without the right to access the information.

A.CLEARANCE: The assumption on the procedures for granting authorization for access to specific security levels is covered by OE.INFO_PROTECT which requires that DAC and MAC protections are set up correctly and that users are trained to perform these tasks properly.

A.SENSITIVITY: The assumption on the procedures for establishing the security level of all information imported to or exported from the system including the security level of peripheral devices is covered by OE.INFO_PROTECT which requires that DAC and MAC protections are set up correctly and that users are trained to perform these tasks properly.

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.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.

P.ACCESS: The access rights to specific objects are determined by O.ROLE which provides role-based access control.

P.CLASSIFICATION: The limitations on access to information based on sensitivity labels are implemented by O.MANDATORY_ACCESS which provides the mandatory access control policy.

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 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 [LSPP]. 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, 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.

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). 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 passwords 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 and FMT_REV.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.

Mandatory Access Control (LSPP mode only)

FDP_IFC.1 requires sensitivity label based mandatory access control for the named objects. The rules enforced are defined in FDP_IFF.2. With mandatory access control active the import and export of both data with its sensitivity labels and data without its sensitivity labels has be performed in accordance with the mandatory access control policy. This is expressed 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. Assignment of initial values for the sensitivity labels when creating a named object is defined in FMT_MSA.3(2). Label exchange is defined in FPT_TDC.1.

Role-based Access Control (LSPP mode only)

FDP_ACC.1(2) and FDP_ACF.1(2) are instantiations that define the role-based access control policy. Roles themselves are defined in FMT_SMR.2. Management of object security attributes is defined by the instantiations FMT_MSA.1(3,4,5,6) and their initial values by FMT_MSA.3(3). Assigning roles to a user upon login is defined in FIA_USB.1.

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 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. 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). FMT_MTD.3 ensures secure values for password selection.

The TOE needs to ensure that appropriate controls are in place for session establishment initiated by users. This is expressed with the security requirements 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 the iterations of FMT_MSA.1 and FMT_REV.1 “Revocation of Object Attributes”). In addition new objects have default access rights which are required by the iterations of FMT_MSA.3.

The second one is the management of users, which is defined in FMT_MTD.1 “Management of User Attributes” and FMT_REV.1 “Revocation of User Attributes”. Since passwords are used for authentication the management of this authentication data is also required in FMT_MTD.1 “Management of Authentication Data” and FMT_MTD.1 “Initialization of Authentication Data”. Management of the audit subsystem is expressed by the requirements for the management of the audit trail (FMT_MTD.1 “Management of the Audit Trail”) and the management of the audit events (FMT_MTD.1 “Management of the Audit Events”). Audit trail management is supported by the requirements for the audit review (FAU_SAR.1, FAU_SAR.2 and FAU_SAR.3) as well as the requirements for the protection of the audit trail (FAU_STG.1, FAU_STG.3 and FAU_STG.4). Management of the audit events is supported by the ability to select the events to be audited (FAU_SEL.1). In addition the TOE supports roles which is expressed by FMT_SMR.2 and FMT_MTD.1 “Management of Roles”.

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.

The configuration of TSF trusted databases is covered by FMT_MSA.3, FMT_MTD.1 (Management of User Attributes; Authentication Data; and Roles), FMT_SMR.2, FMT_SMF.1, and FMT_MTD.3.

The TOE also needs to provide tools that allow the administrator to check the integrity of the underlying hardware and the correct operation of the TSF. Such abilities are addressed by FPT_AMT.1 and FPT_TST.1.

The TOE needs to enter a secure state when critical security functions fail, and allow the administrator to perform repairs and re-enter the normal operating mode. This is expressed with the security requirements FPT_FLS.1, FPT_RCV.1, FPT_RCV.4, and FPT_RVM.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) Secure security attributes (FMT_MSA.2) Management of Authentication Data (FMT_MTD.1) 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	Management of Object Security Attributes (FMT_MSA.1(1,2,3,4,5,6)) Static Attribute Initialization (FMT_MSA.3(1,2,3)) Management of the Audit Trail (FMT_MTD.1) Management of Audited Events (FMT_MTD.1) Management of User Attributes (FMT_MTD.1) Initialization of Authentication Data (FMT_MTD.1) Management of Authentication Data (FMT_MTD.1) Management of Roles (FMT_MTD.1) Revocation of User Attributes (FMT_REV.1) Management of Object Security Attributes (FMT_MSA.1(1,2,3,4,5,6)) Revocation of Object attributes (FMT_REV.1) Specification of Management Functions (FMT_SMF.1) Security Management Roles (FMT_SMR.2) Manual Recovery (FPT_RCV.1) Function Recovery (FPT_RCV.4)
O.ENFORCEMENT	Reference Mediation (FPT_RVM.1) Domain Separation (FPT_SEP.1) Abstract Machine Testing (FPT_AMT.1) Fail secure (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)

Objective	Security Functional Requirement
	Management of the Audit Trail (FMT_MTD.1) Management of Audited Events (FMT_MTD.1) Reliable Time Stamps (FPT_STM.1) Security Roles (FMT_SMR.2)
O.COMPROT	Cryptographic Key Generation (FCS_CKM.1 (1-4)) Cryptographic Key Distribution (FCS_CKM.2 (1-5)) Cryptographic Operation (FCS_COP.1 (1-4)) 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) Management of Object Security Attributes (FMT_MSA.1(2)) Revocation of Object Attributes (FMT_REV.1) Static Attribute Initialization (FMT_MSA.3(2)) Inter-TSF TSF data consistency (FPT_TDC.1)
O.DUTY	Security Roles (FMT_SMR.2)
O.HIERARCHICAL	Management of Roles (FMT_MTD.1)
O.ROLE	Security Roles (FMT_SMR.2) 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(3,4,5,6)) Static Attribute Initialization (FMT_MSA.3(3)) 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, FMT_MSA.2]. 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 [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 enforcement of role separation [FMT_SMR.2] ensures that audit data can be reliably 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 [FMT_MTD.1 “Management of the Audit Trail” and FMT_MTD.1 “Management of the Audit Events”] and user accounts [FMT_MTD.1 "Management of User Attributes", FMT_MTD.1 "Management of Authentication Data", FMT_REV.1 "Revocation of User Attributes"] and object attributes [FMT_MSA.1, FMT_REV.1 "Revocation of Object Attributes"]. In addition the default values for access control need to be defined [FMT_MSA.3]. A mechanism for exchange of labeled data among systems must be defined [FPT_TDC.1].

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 [FPT_AMT.1], a self test utility [FPT_TST.1] and support entering a fail secure mode on critical errors [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 must control access to resources based on the sensitivity labels of subjects and objects. The TSF must allow authorized users to specify which resources may be accessed by which users.

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, FPT_TDC.1].

Mandatory access control must have a defined scope of control [FDP_IFC.1]. The rules of the MAC policy must be defined [FDP_IFF.1]. The security attributes of objects used to enforce the MAC policy must be defined. The security attributes of subjects used to enforce the MAC 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(2)] 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(2)].

O.DUTY

The TOE must provide the capability of enforcing ‘separation of duties’. The enforcement of role separation [FMT_SMR.2] supports this objective.

O.HIERARCHICAL

The TOE must provide the capability of defining hierarchical roles as required by [FMT_MTD.1].

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(3,4,5,6)] 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.

Security requirements for the IT environment have been added to define the support required by the TOE from the underlying processor. As with every operating system that also runs untrusted software, some kind of separation mechanism must exists that prohibits the untrusted software from tampering with trusted software and TSF data. In the case of this TOE the processor must supply a separation mechanism such that memory areas as well as hardware privileges required to directly access devices or memory management functions are protected from direct access by untrusted software. This is defined with an access control policy called „memory access control policy" that the underlying processor must support. This policy is expressed using FDP_ACC.1 and FDP_ACF.1 as well as FDP_MSA.3 from part 2 of the Common Criteria.

8.2.2 Security Requirements Coverage

The following table shows that each security functional requirement addresses at least one objective.

Table 8-7: Mapping Security Functional Requirements to Objectives

SFR	Objectives
FAU_GEN.1	O.AUDITING
FAU_GEN.1	O.AUDITING
FAU_SAR.1	O.AUDITING
FAU_SAR.2	O.AUDITING
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_ACF.1(1)	O.DISCRETIONARY_ACCESS
FDP_ACC.1(2)	O.ROLE
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,

SFR	Objectives
	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, O.MANAGE
FMT_MSA.1(2)	O.MANDATORY_ACCESS, O.MANAGE
FMT_MSA.1(3)	O.ROLE, O.MANAGE
FMT_MSA.1(4)	O.ROLE, O.MANAGE
FMT_MSA.1(5)	O.ROLE, O.MANAGE
FMT_MSA.1(6)	O.ROLE, O.MANAGE
FMT_MSA.2	O.AUTHORIZATION, O.COMPROT
FMT_MSA.3(1)	O.DISCRETIONARY_ACCESS, O.MANAGE
FMT_MSA.3(2)	O.MANDATORY_ACCESS, O.MANAGE
FMT_MSA.3(3)	O.ROLE, O.MANAGE
FMT_MTD.1 Audit Trail	O.AUDITING, O.MANAGE
FMT_MTD.1 Audited Events	O.AUDITING, O.MANAGE
FMT_MTD.1 User Attributes	O.MANAGE
FMT_MTD.1 Authentication Data	O.AUTHORIZATION, O.MANAGE
FMT_MTD.1 Initialization of Authentication Data	O.MANAGE
FMT_MTD.1 Management of Roles	O.MANAGE, O.HIERARCHICAL
FMT_MTD.3	O.AUTHORIZATION
FMT_REV.1 User Attributes	O.MANAGE
FMT_REV.1 Object Attributes	O.DISCRETIONARY_ACCESS, O.MANDATORY_ACCESS, O.ROLE, O.MANAGE
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.ROLE
FTA_TSE.1	O.ROLE
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 8-9: 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(1)	FDP_ACF.1 Security attribute based access control	Yes
FDP_ACF.1(1)	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	Yes
FDP_ACC.1(2)	FDP_ACF.1 Security attribute based access control	Yes
FDP_ACF.1(2)	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	Yes
FDP_ETC.1	FDP_IFC.1 Subset information flow control	Yes
FDP_ETC.2	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	Yes
FDP_ITC.1	[FDP_ACC.1 or FDP_IFC.1] FMT_MSA.3	Yes
FDP_ITC.2	[FDP_ACC.1, or FDP_IFC.1] [FTP_ITC.1, or FTP_TRP.1] FPT_TDC.1	Yes
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)

Security Functional Requirement	Dependencies	Resolved
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.1 Security roles FMT_SMF.1Specification of management function	Yes
FMT_MSA.2	ADV_SPM.1 Informal TOE security Policy model [FDP_ACC.1 Subset access control or FDP_IFC.1 Subset infromation flow control] FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	Yes
FMT_MSA.3	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles FMT_SMF.1 1Specification of management function	Yes
FMT_MTD.1 Audit Trail	FMT_SMR.1 Security Roles	Yes
FMT_MTD.1 Audit Events	FMT_SMR.1 Security Roles	Yes
FMT_MTD.1 User Attributes	FMT_SMR.1 Security roles	Yes
FMT_MTD.1 Authentication Data	FMT_SMR.1 Security roles	Yes
FMT_MTD.1 Initialization of Authentication Data	FMT_SMR.1 Security roles	Yes
FMT_MTD.1 Management of Roles	FMT_SMR.1 Security roles	Yes
FMT_MTD.3	ADV_SPM.1 FMT_MTD.1	Yes
FMT_REV.1 User Attributes	FMT_SMR.1 Security roles	Yes
FMT_REV.1 Object Attributes	FMT_SMR.1 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	Yes
FPT_RCV.1	AGD_ADM.1 ADV_SPM.1	Yes
FPT_RCV.4	ADV_SPM.1	Yes

Security Functional Requirement	Dependencies	Resolved
FPT_RVM.1	No dependencies	Yes
FPT_SEP.1	No dependencies	Yes
FPT_STM.1	No dependencies	Yes
FPT_TDC.1	No dependencies	Yes
FPT_TST.1	FPT_AMT.1	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 on FIA_UID.1 are resolved with the inclusion of FIA_UID.2 which is hierarchical to FIA_UID.1

The dependencies on FMT_SMR.1 are resolved with the inclusion of FMT_SMR.2 which is hierarchical to FMT_SMR.1.

The dependencies on FDP_IFF.1 are resolved with the inclusion of FDP_IFF.2 which is hierarchical to FDP_IFF.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 FDP_ACC.1, FDP_ACF.1, FMT_MSA.1, FMT_MSA.3, 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 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.

8.2.5 Evaluation Assurance Level

This security target claims EAL4 augmented with ALC_FLR.3, which is seen appropriate for a controlled environment where attackers only have a low attack potential.

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 8-10: 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**.
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	The multiple instantiations of this security functional requirement are described in **SC** where the SSH v2.0 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	The multiple instantiations of this security functional requirement are described in **SC** where the SSH v2.0 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	The multiple instantiations of this security functional requirement are described in **SC** where the SSH v2.0 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.
FDP_ACF.1(1)	The discretionary access control is realized as described above by **DA**. There the individual mechanisms for access control depending on the object type are described in detail.
FDP_ACC.1(2)	The role-based access control policy is based on **RA** defining the subjects and objects covered by this policy.
FDP_ACF.1(2)	The role-based access control is realized as described above by **RA**. There the individual mechanisms for access control depending on the object type are described in detail.
FDP_ETC.1	The export of unlabeled data is covered by the MAC policy described in **MA**
FDP_ETC.2	The export of labeled data is covered by the MAC policy described in **MA**
FDP_IFC.1	The mandatory access control policy is based on **MA** defining sensitivity labels for the subjects and objects.
FDP_IFF.2	The mandatory access control is realized as described above by **MA**. There the individual mechanisms for access control depending on the object type are described in detail.
FDP_ITC.1	The import of unlabeled data is covered by the MAC policy described in **MA**
FDP_ITC.2	The import of labeled data is covered by the MAC policy described in **MA**
FDP_RIP.2	Object residual information protection is realized by security functions for

SFR	Security Functions (TOE Summary Specification)
	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 v2.0 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 v2.0 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**.
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 enforcement of the user-subject binding is covered by **DA**, **MA**, and **RA**.
FMT_MSA.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), **MA**, and **RA**.
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.
FMT_MSA.3	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**, **DA**, **MA**, and **RA**). Some default values are defined in TSF databases as defined in **TP**.
FMT_MTD.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 Audited Events	The way an authorized administrator can select the events to be audited is defined in **AU** and **SM**.
FMT_MTD.1 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 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 Initialization of Authentication Data	Initialization of authentication data is restricted to administrative users during the creation of new users in **SM**.
FMT_MTD.1 Management of Roles	Management of roles is restricted to administrators in **SM**.
FMT_REV.1 User Attributes	The revocation of user security attributes as required in FMT_REV.1 is realized by the user management functions of **SM** and enforced by **DA**,

SFR	Security Functions (TOE Summary Specification)
	MA, and **RA**.
FMT_REV.1 Object Attributes	Revocation of object security attributes is realized by the access control configuration and management function **SM** and enforced by **DA**, **MA**, and **RA**.
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**. 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 functions **SM** and **RA**.
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	The operation of the secure failure state is covered by the function **SM**.
FPT_RCV.1	The recovery from a secure failure state is covered by the function **SM**.
FPT_RCV.4	The recovery from a secure failure state is covered by the function **SM**.
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	The exchange of labeled data is described in **MA.**
FPT_TST.1	The ability of the authorized administrator to test the functions of the TSF are described in **TP**.
FTA_LSA.1	The restrictions on initiating user sessions are covered by the **IA** function.
FTA_TSE.1	The restrictions on initiating user sessions are covered by the **IA** function.
FTP_ITC.1	The function for setting up a trusted channel between the TOE and another trusted IT product using the SSH v2.0 or SSL v3 protocol is described in **SC.**

This table shows how the security functions work together to satisfy the security functional requirements.

Access control is defined by a discretionary and role-based access control policy in FDP_ACC.1(1,2) and FDP_ACF.1(1,2) and a mandatory access control policy in FDP_ETC.1/2, FDP_IFC.1, FDP_IFF.2, FDP_ITC.1/2, and FPT_TDC.1. A security domain is enforced by restricting access to security relevant objects to authorized users as stated in FPT_SEP.1. For Red Hat Enterprise Linux there are two different types of objects with some differences in policies depending on the object type. All the dependencies on the management aspects have been resolved. The management of the two object types differs only slightly, where those differences are explained in FMT_MSA.1 and FMT_REV.1.

Audit of events is performed to be able to hold users accountable for their activities. Generation of audit records including the login ID of the user is addressed by FAU_GEN.1 and FAU_GEN.2. The availability of the audit trail is addressed by FAU_STG.1, FAU_STG.3, and FAU_STG.4. The audit trail must be secured from unauthorized access as described in FAU_SAR.2. Review of the audit trail by the administrator is discussed in FAU_SAR.1 and FAU_SAR.3. The management of both the audit trail and the audited events is described in FMT_MTD.1 and FAU_SEL.1.

Object reuse is a useful requirement to prohibit unwanted access to information via resources that have not been prepared for reuse. Since the TOE supports access control, object reuse makes sense. This is addressed in FDP_RIP.2.

Secure communication is used to protect data in transit between the TOE and trusted IT against disclosure and undetected unauthorized modifications as described in FDP_UCT.1 and FDP_UIT.1. There needs to be a trusted channel between the TOE and other trusted IT as defined in FTP_ITC.1. The generation of cryptographic keys for the mechanisms involved is addressed by FCS_CKM.1; the distribution of such keys is discussed in FCS_CKM.2. The cryptographic algorithms used are detailed in FCS_COP.1. As described in FMT_MSA.2 only secure values are allowed for cryptographic keys.

Identification and authentication is handled by FIA_ATD.1, FIA_SOS.1 FIA_UAU.2, FIA_UAU.7, FIA_UID.2, FIA_USB.1, FTA_LSA.1, and FTA_TSE.1 in a fairly conventional way. FIA_USB describes the way the effective and filesystem user ID and group ID can be changed.

In the management section the requirements for the management User Attributes, Authentication Data, Roles, and Audit Configuration has been separated in this Security Target. Since they are clearly separated, they are not contradicting each other.

Revocation for user attributes is described separately from revocation of object attributes in two instantiations of FMT_REV.1. This makes sense, since revocation is handled differently. FMT_SMF.1 has been included because of [CC] and covers the different management aspects addressed in detail in FMT_MSA.1 and the instantiations of FMT_MTD.1.

The TOE supports roles as expressed by FMT_SMR.2.

FPT_RVM.1 is required to ensure that the security functions can not be bypassed. In addition FPT_SEP.1 ensures that untrusted programs can not tamper with the TSF and cause them to operate in contradiction to the security policy of the TOE. Failures of critical security functions results in a secure state (FPT_FLS.1) and a method to recover from it (FPT_RCV.1/4). FPT_AMT.1, FPT_TST.1, FPT_RVM.1 and FPT_SEP.1 are therefore mutually supportive requirements to enable a sufficient self-protection of the TSF.

As a summary this shows that the security functional requirements are not contradicting each other and are mutually supportive.

8.3.2 Assurance Measures Justification

The TOE summary specification in section 6.4 of this document 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 of this document 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 of this document.

The SOF-medium claim does **not** apply to the cryptographic algorithms, including the cryptographic properties of the 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 412.

Therefore, a strength of SOF-medium is consistent with the description of the TOE environment.

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 Organization for Standardization
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
TSC	TSF Scope of Control (the set of interactions that can occur with or within a TOE and are subject to the rules of the TSP)
TSF	TOE Security Functions
UDP	User Datagram Protocol
VFS	Virtual File System
VMM	Virtual Memory Manager