WIND RIVER

Wind River Linux Secure 1.0 Security Target

Version:	**1.17**
Status:	**Released**
Last Update:	**2011-04-06**
Classification:	**Public**

Trademarks

Wind River and the Wind River logo are trademarks or registered trademarks of Wind River Systems, Inc. in the United States, other countries, or both.

atsec is a trademark of atsec information security GmbH

Linux is a registered trademark of Linus Torvalds.

UNIX is a registered trademark of The Open Group in the United States and other countries.

IBM, IBM logo, bladecenter, eServer, iSeries, OS/400, , POWER3, POWER4, POWER4+, pSeries, System p, POWER5, POWER5+, System x, System z, S390, xSeries, zSeries, zArchitecture, and z/VM are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

Intel, Xeon, and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

Legal Notice

This document is provided AS IS with no express or implied warranties. Use the information in this document at your own risk.

This document may be reproduced or distributed in any form without prior permission provided the copyright notice is retained on all copies. Modified versions of this document may be freely distributed provided that they are clearly identified as such, and this copyright is included intact.

Table of Contents

List of Tables

1 Introduction

1.1 Security Target Identification

Title:	Wind River Linux Secure 1.0 Security Target
Version:	1.17
Status:	Released
Date:	2011-04-06
Sponsor:	Wind River Systems, Inc.
Developer:	Wind River Systems, Inc.
Validation ID:	10430
Keywords:	Security Target, Common Criteria, Linux Distribution, Embedded Linux

1.2 TOE Identification

The TOE is Wind River Linux Secure Version 1.0.

1.3 TOE Type

The TOE type is Linux-based operating system intended for embedded devices.

1.4 TOE Overview

This security target documents the security characteristics of the Wind River Linux Secure 1.0 distribution which is derived from the Wind River Linux 3.0.3 distribution. Please note that the TOE version is a revision level release of the 3.0 release which implements all the described mechanisms in this ST.

Wind River Linux Secure 1.0 is a commercial-grade Linux solution for embedded device development. The platform contains a fully tested, validated, and supported Linux distribution based on the Linux 2.6.27 kernel technology.

Wind River Linux meets the demands of embedded device developers for markets such as the aerospace and defense, networking, industrial and medical devices, and consumer electronics.

1.4.1 Required Hardware and Software

The hardware / firmware component of the TOE which allows the installation of the operating system on the following hardware systems:

- Dell D630 (using Intel Core 2 Duo processor)
- Intel 'Hanlan Creek' Dual Processor Xeon 5500 Series Pedestal Server Motherboard (S5520HCR) (using Intel Nehalem processor)
- PPC_32 MPC8572DS (using Freescale MPC8572 PowerPC 32 bit processor)
- ARM TI OMAP3530 (using ARM Cortex-A8 processor)
- SolCORE ITAR-restricted board

The listed boards have a form factor, physical interfaces and a power consumption that supports the use as embedded devices. There are no specialized hardware devices inside the TSF, nor that are available to be accessed by the user in any way.

Wind River Linux Secure provides a specialized installation procedure to support embedded systems. Using this installation system, installation images are generated which in turn are copied onto the disk device of the target board.

In addition, Wind River Linux Secure allows developers to analyze the system and his applications with extensive and specialized debugging features, including the Linux Trace Toolkit. Note that these debugging features must be disabled in the evaluated configuration.

1.4.2 Intended Method of Use

All human users, if existent, as well as all services offered by the embedded system are assigned unique user identifiers within the single host system that forms the TOE. This user identifier is used together with the attributes and roles assigned to the user identifier 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. Services may be spawned by the TOE without the need for user-interaction. The TOE internally maintains a set of identifiers associated with processes, which are derived from the unique user identifier upon login of the user or from the configured user identifier for a TOE-spawned service. Some of those identifiers may change during the execution of the process according to a policy implemented by the TOE.

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

The TOE uses mandatory access control together with discretionary access control. Rules are defined to assign sensitivity labels to subjects and objects and to implement the information flow mandatory access control policy based on the Bell-LaPadula model.

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 is capable of securely allocating resources to multiple users of the TOE. These resources include multiple processors, memory, and attached peripheral and storage devices. The TOE facilitates controlled sharing of these resources based on subject and object security attributes.

Many processes that are run on the TOE automatically without requiring user interaction do not have full privilege to the system. Although they are not bound to users, they are still subject to access control. The TOE is designed this way to enforce the principle of least privilege. Such installations and usage scenarios are typical for embedded systems that are accessed predominantly by other technical entities.

It is assumed that responsibility for the safeguarding of the data protected by the TOE can be delegated to human users of the TOE if such users are allowed to log on and spawn processes on their behalf. For embedded systems, users are typically not allowed to log on to the system but different UIDs are used to separate different services provided by the embedded system. In such a case, it is assumed that these processes are responsible for the safeguarding of their data. All data is under the control of the TOE. The data is stored in named objects, and the TOE can associate a description of the access rights to that object with each named object.

Note: An embedded system provides the platform for installing and running arbitrary services. These additional services are not part of the TOE. The TOE is solely the operating system which provides the runtime environment for such services.

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, and by administrative users. 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 identified with their UID. 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.

1.4.3 Major Security Features

The primary security features of the TOE are:

- Identification and Authentication
- Audit
- Discretionary Access Control
- Mandatory Access Control
- Cryptographic services
- Security Management
- 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.

1.5 TOE Description

1.5.1 Introduction

Wind River Linux is a general purpose, multi-user, multi-tasking Linux based operating system intended for embedded devices. It provides a platform for a variety of applications in the embedded environment. Wind River Linux is available on a broad range of CPUs and associated system boards.

The SELinux security module is configured to enforce the mandatory access control policy based on the labels of subjects and objects using the Bell-LaPadula access control model as a basis.

The Wind River Linux evaluation covers a potentially distributed, but closed network of systems running the evaluated versions and configurations of Wind River Linux as well as other well-behaved peer systems operating within the same management domain. 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 Wind River 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 Wind River Linux are considered to be part of the TOE.

The TOE includes standard networking applications, such as SSH.

System administration tools include the standard command line tools. 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.

1.5.2 TOE boundaries

1.5.2.1 Physical

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

- Wind River Linux in the above mentioned version

The TOE and its documentation are supplied on CD-ROM. The TOE includes a package holding the additional user and administrator documentation.

In addition to the installation media, the following documentation is provided:

- Evaluated Configuration Guide [ECG] - note that this guide is the main guide covering the evaluated configuration settings and requirements;
- Wind River Linux Secure Administrator's Guide [WRLSAG] ;
- Wind River Linux Secure Configuration Guide [WRLSCG] .

The hardware that is applicable to the evaluated configuration is listed in 1.4.1 . The analysis of the hardware capabilities as well as the firmware functionality is covered by this evaluation to the extent that the following capabilities supporting the security functionality are analyzed and tested:

- Memory separation capability
- Unavailability of privileged processor states to untrusted user code (like the hypervisor state or the SMM)
- Full testing of the security functionality on all listed boards

1.5.2.2 Logical

The primary security features of the TOE are:

- Identification and Authentication: User identification and authentication in the TOE includes all forms of interactive login (e.g. using the SSH protocol or log in at the local console) as well as identity changes through the su command. These all rely on explicit authentication information provided interactively by a user.
- Audit: The Lightweight Audit Framework (LAF) is designed to be an audit system for Linux compliant with the requirements from Common Criteria. LAF is able to intercept all system calls as well as retrieving audit log entries from privileged user space applications. The subsystem allows configuring the events to be actually audited from the set of all events that are possible to be audited.
- Discretionary Access Control: DAC allows owners of named objects to control the access permissions to these objects. These owners can permit or deny access for other users based on the configured permission settings.

- Mandatory Access Control: The TOE supports mandatory access control using sensitivity labels automatically attached to processes and objects. Users cannot interfere with these labels. The access control policy enforced using these labels is derived from the Bell-LaPadula access control model. The TOE uses SELinux with an appropriate policy to enforce the mandatory access control.
- Cryptographic services: The TOE provides the NSS library which is covered by a FIPS 140-2 certificate. The NSS library is used in a FIPS 140-2 compliant mode for generic cryptographic services and integrity checking of the TSF. The cryptographic mechanisms the security functionality of this document relies on is validated according to FIPS 140-2, certificate number 1475 (non-ITAR platforms), and 1506 (ITAR platform).
- Security Management: The security management facilities provided by the TOE are usable by authorized users and/or authorized administrators to modify the configuration of TSF.
- TSF Protection: The TSF is structured such that the TSF has exclusive access to the system's resources. Access requests to these resources by users must be mediated by the TSF. Various additional protection mechanisms are in place to avoid misuse of the TOE.

The TOE provides many more functions and mechanisms. The evaluation ensures that all these additional functions do not interfere with the above mentioned security mechanisms in the evaluated configuration. Mechanisms and functions that would interfere with the operation of the security functions are disallowed in the evaluated configuration and the Evaluation Configuration Guide provides instructions to the administrator on how to disable them. Note: TOE mechanism which provide additional restrictions to the above claimed security functions are allowed in the evaluated configuration. For example, BSDJails are provided with the TOE and permitted in the evaluated configuration even though they have not been subject to this evaluation. BSDJails provide futher restrictions on, for example, the security function of discretionary access control mechanism for IPC objects and therefore cannot breach the security functionality. The following table enumerates mechanisms that are provided with the TOE but which are excluded from the evaluation (note that if a function is marked as "unavailable" it is not active or not present in the TOE, if a function is marked as "not assessed" it is active but the evaluation did not analyze it an no security claims apply to it):

Functionalities	**Exclusion discussion**
OpenSSL, beecrypt, gnutls	The OpenSSL library, the beecrypt library as well as the gnutls library are not intended to provide the cryptographic functionality claimed by this ST and are therefore not assessed. Users and administrators are provided with guidance on how to use the NSS functionality instead. The correct implementation of the ciphers provided with the mentioned libraries is asserted by the vendor.
Cryptography provided with gnupg, duplicity	The applications of gnupg and duplicity provide cryptography which is not covered by this evaluation.
OpenSSH	The cryptographic aspects of OpenSSH are not assessed.
eCryptFS	eCryptFS is not allowed to be used in the evaluated configuration. The encryption capability provided with this file system is therefore unavailable to any user.
GRSecurity	The functionality offered by GRSecurity is not assessed as part of the evaluation.

Functionalities	Exclusion discussion
PaX	The functionality of preventing the exploitation of software bugs is not assessed as part of the evaluation.
SMACK	The mandatory access control functionality offered by the SMACK LSM is unavailable in the evaluated configuration. The SELinux LSM provides the mandatory access control policy enforcement.
IPSEC	IPSEC must be used with the TOE, but the cryptographic aspects of IPSEC are not assessed. The IPSEC tunnel is only used to provide a communication channel which is capable of transporting labeled data.
SSL / TLS tunnels	The TOE provides the stunnel application which can be used to establish SSL and TLS tunnels with remote peers. This application however is not assessed.
Linux kernel packet filtering	The packet filter functionality provided by the Linux kernel with the netfilter / iptables mechanism is not assessed in this evaluation.
KVM	The KVM virtualization mechanism is available in the TOE but is not assessed in this evaluation.
BSDJails	As mentioned above, the BSDJails are not assessed in this evaluation.
SELinux type enforcement	The type enforcement policy (including the RBAC mechanism) provided with SELinux is not assessed with this evaluation.
Printing support not available	The evaluated configuration does not provide any printer support, including the cups printing server.
Mail server functionality not available	The evaluated configuration does not provide any mail server functionality, including distribution of results from cron jobs due to MLS restrictions.
Key retention services	The key retention service mechanism of the Linux kernel is not active.
aide, samhain integrity check	The integrity checking mechanism provided with aide or samhain is not covered in this evaluation.
Kernel functionality: ext2, configfs, NFS, NUMA	Kernel functions that are not enabled in the evaluated configuration are not subject to this evaluation. Among others, the ext2 file system, the configfs file system, the NFS file system, or the NUMA memory management functions are not covered.
Applications: rnano, vsftpd, libc crypt function	Applications that are either not enabled in the evaluated configuration or do not provide the capability for users to elevate their privileges are not subject to this evaluation. Among others, the rnano editor, the vsftpd FTP server, or the libc crypt() function are not covered.

Table 1: Non-evaluated functionalities

Cryptography in the product that is not related to the security functional requirements is not covered by the evaluation and their validity is vendor asserted. This applies to all applications not linked to the FIPS 140-2 validated NSS library provided with the Wind River Cryptographic framework. Examples identified during the evaluation include: OpenSSL, beecrypt, gnutls, gnupg, duplicity, OpenSSH, eCryptFS, IPSec, stunnel, samhain, and the crypt() function.

Note: The exclusion of the above packages and mechanisms from this evaluation have been excluded due to resource constraints. Their exclusion does not imply the packages are insecurely implemented. However, since they are not evaluated, administrators are advised to use them at their own risk.

Note: Compliant with the chosen EAL, no formal covert channel analysis has been performed.

1.5.2.3 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.
- Wind River Linux supports the use of IPv4 and IPv6, both are also supported in the evaluated configuration except for the ITAR-restricted board.
- The default configuration for identification and authentication are the defined password-based PAM modules. Support for other authentication options, e.g. smart card 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.

Deviations from the configurations and settings specified with the Evaluated Configuration Guide are not permitted.

The TOE comprises a single embedded system (and optional peripherals) running the TOE software listed.

The TOE in the evaluated configuration comprises of the packages in the following table. Note that while some packages are included in the TOE, they may not be used or were not evaluated. For further information, see section 1.5.2.2 and [ECG] for further details:

acl-2.2.39-1.1_WR1.0.0ao.x86_64
acpid-1.0.6-7_WR_1.0.0ao.2.x86_64
adduser-3.110-1_WR1.0.0ao.x86_64
at-3.1.10-11_WR1.0.0ao.x86_64
attr-2.4.43-1_WRS1.0.0ao.x86_64
audit-1.7.12-4_WR1.0.0ao.x86_64
audit-libs-1.7.12-4_WR1.0.0ao.x86_64
audit-test-2090-1_WRS1.0.0ao.x86_64
bash-3.2-22_WR1.0.0ao.x86_64
beecrypt-4.1.2-12_WR1.0.0ao.x86_64
bzip2-1.0.5-1_WR1.0.0ao.x86_64

bzip2-libs-1.0.5-1_WR1.0.0ao.x86_64
checkpolicy-2.0.20-1_WRS1.0.0ao.x86_64
chkconfig-1.3.34-1_WR1.0.0ao.x86_64
common_pc_64-config-1.0-1_WRS1.0.0ao.x86_64
common_pc_64-kernel-2.6.27.47-1_WRS1.0.0ao.x86_64
coreutils-6.9-2_WR1.0.0ao.x86_64
cpio-2.6-27_WR1.0.0ao.1.x86_64
cracklib-2.8.9-6_WR1.0.0ao.x86_64
cracklib-dicts-2.8.9-6_WR1.0.0ao.x86_64
cracklib-python-2.8.9-6_WR1.0.0ao.x86_64
crontabs-1.10-19_WR1.0.0ao.noarch
curl-7.19.3-1_WR1.0.0ao.1.x86_64
daemontools-0.76-6rph_WR1.0.0ao.x86_64
db4-4.6.21-5_WR1.0.0ao.x86_64
db4-utils-4.6.21-5_WR1.0.0ao.x86_64
device-mapper-1.02.19-1_WR1.0.0ao.x86_64
device-mapper-libs-1.02.19-1_WR1.0.0ao.x86_64
diffutils-2.8.1-21_WR_1.0.0ao.x86_64
duplicity-0.6.08b-_WR1.0.0ao.x86_64
e2fsprogs-1.40.8-2_WR1.0.0ao.x86_64
e2fsprogs-libs-1.40.8-2_WR1.0.0ao.x86_64
ed-0.5-1_WR1.0.0ao.x86_64
elfutils-0.108_1-1_WRS1.0.0ao.x86_64
ethtool-6-1_WR1.0.0ao.x86_64
eventlog-0.2.5-8_WR1.0.0ao.x86_64
evlog-1.6.1-1_WR1.0.0ao.x86_64
expat-2.0.1-5_WR1.0.0ao.x86_64
expect-5.43.0-12_WR1.0.0ao.x86_64
fam-2.7.0-1_WRS1.0.0ao.x86_64
file-4.23-5_WR1.0.0ao.x86_64
file-libs-4.23-5_WR1.0.0ao.x86_64

filesystem-2.4.13-1_WR1.0.0ao.x86_64
findutils-4.2.31-3_WR1.0.0ao.1.x86_64
gawk-3.1.5-15_WR1.0.0ao.1.x86_64
gdbm-1.8.3-1_WRS1.0.0ao.x86_64
gettext-0.16.1-8_WR1.0.0ao.x86_64
glib2-2.16.3-5_WR1.0.0ao.x86_64
glibc-2.8-1_WR4.3a_274.0.0.0.0.x86_32
glibc-2.8-1_WR4.3a_274.0.0.0.0.x86_64
glibc-common-2.8-1_WR4.3a_274.0.0.0.0.x86_32
glibc-common-2.8-1_WR4.3a_274.0.0.0.0.x86_64
glibc-locale-2.8-1_WR4.3a_274.0.0.0.0.x86_64
gmp-4.2.1-3_WR1.0.0ao.1.x86_64
gnupg2-2.0.4-1_WR1.0.0ao.x86_64
gradm-2.1.12-200812271437_1_WR1.0.0ao.x86_64
grep-2.5.1-57_WR1.0.0ao.1.x86_64
grub-0.97-1_WRS1.0.0ao.x86_32
gzip-1.3.12-3_WR1.0.0ao.2.x86_64
heartbeat-2.1.3-2_WR1.0.0ao.x86_64
ifenslave-1.1.0-1_WRS1.0.0ao.x86_64
inetutils-1.4.2-1_WRS1.0.0ao.x86_64
initscripts-8.76-1_WR1.0.0ao.x86_64
inotify-tools-3.13-1_WRS1.0.0ao.x86_64
iproute-2.6.20-2_WR1.0.0ao.x86_64
ipsec-tools-0.7-13_WR1.0.0ao.x86_64
iptables-1.4.3.1-1_WR1.0.0ao.x86_64
iptables-ipv6-1.4.3.1-1_WR1.0.0ao.x86_64
iputils-20071127-2_WR1.0.0ao.x86_64
kexec-tools-2.0_rc-20080318_WR1.0.0ao.1.x86_64
keynote-2.3-1_WRS1.0.0ao.x86_64
krb5-libs-1.6-3_WR1.0.0ao.5.x86_64
krb5-server-1.6-3_WR1.0.0ao.5.x86_64

krb5-workstation-1.6-3_WR1.0.0ao.5.x86_64
less-394-9_WR1.0.0ao.x86_64
libacl-2.2.39-1.1_WR1.0.0ao.x86_64
libaio-0.3.106-4.2_WR1.0.0ao.x86_64
libassuan-devel-1.0.1-1_WR1.0.0ao.1.x86_64
libcap-2.10-2_WR1.0.0ao.x86_64
libcurl4-7.19.3-1_WR1.0.0ao.1.x86_64
libevent-1.4.6_stable-1_WRS1.0.0ao.x86_64
libgcc-4.3a_274-1_WR4.3a_274.0.0.0.0.x86_32
:libgcc-4.3a_274-1_WR4.3a_274.0.0.0.0.x86_64
libgcrypt-1.4.0-3_WR1.0.0ao.x86_64
libgpg-error-1.6-2_WR1.0.0ao.x86_64
libidn-0.6.5-1.1_WR1.0.0ao.x86_64
libksba-1.0.1-1_WRS1.0.0ao.x86_64
libnl-1.1-3_WR1.0.0ao.1.x86_64
libpcap-0.9.8-2_WR1.0.0ao.x86_64
librsync-0.9.7-10.x86_64
libselinux-2.0.89-1_WRS1.0.0ao.x86_64
libsemanage-2.0.42-1_WRS1.0.0ao.x86_64
libsepol-2.0.41-1_WRS1.0.0ao.x86_64
libstdc++-4.3a_274-1_WR4.3a_274.0.0.0.0.x86_64
libsysfs-2.1.0-3_WR1.0.0ao.x86_64
libtool-2.2.4-1_WRS1.0.0ao.x86_64
libusb-0.1.12-15_WR_1.0.0ao.x86_64
libvolume_id-120-5.20080421git_WR1.0.0ao.x86_64
libxml2-2.6.30-1_WR1.0.0ao.x86_64
lm_sensors-3.0.1-5_WR_1.0.0ao.x86_64
logcheck-1.1.1-1_WRS1.0.0ao.x86_64
logrotate-3.7.4-7_WR1.0.0ao.x86_64
lsof-4.78-5_WR1.0.0ao.x86_64
lspp-eal4-config-ibm-0.65-2_WR1.0.0ao.noarch

ltp-full-20090531-1_WRS1.0.0ao.3.x86_64
lvm2-2.02.25-1_WRS1.0.0ao.x86_64
mailx-8.1.1-44.2.2_WR1.0.0ao.x86_64
MAKEDEV-3.23-1.2_WR1.0.0ao.x86_64
mcelog-0.7-1.23_WR1.0.0ao.x86_64
mcstrans-0.2.5-1_WR1.0.0ao.x86_64
mdadm-2.6.1-4_WR1.0.0ao.x86_64
mhash-0.9.9-1_WRS1.0.0ao.x86_64
mingetty-1.08-1_WRS1.0.0ao.x86_64
minicom-2.3-2_WR_1.0.0ao.x86_64
mipv6-daemon-umip-0.3-2_WR1.0.0ao.x86_64
mktemp-1.5-25_WR1.0.0ao.x86_64
mm-1.4.2-4.x86_64
module-init-tools-3.2.2-1_WRS1.0.0ao.x86_64
mtree-2.7-0._WR.x86_64
ncftp-3.2.3-2.3.x86_64
ncurses-5.6-19.20080628_WR1.0.0ao.x86_32
ncurses-5.6-19.20080628_WR1.0.0ao.x86_64
ncurses-base-5.6-19.20080628_WR1.0.0ao.x86_32
ncurses-base-5.6-19.20080628_WR1.0.0ao.x86_64
ncurses-libs-5.6-19.20080628_WR1.0.0ao.x86_32
ncurses-libs-5.6-19.20080628_WR1.0.0ao.x86_64
ncurses-term-5.6-19.20080628_WR1.0.0ao.x86_32
ncurses-term-5.6-19.20080628_WR1.0.0ao.x86_64
neon-0.28.2-1_WRS1.0.0ao.x86_64
netcat-1.10-1_WRS1.0.0ao.x86_64
net-snmp-5.4-13_WR1.0.0ao.x86_64
net-snmp-libs-5.4-13_WR1.0.0ao.x86_64
net-snmp-utils-5.4-13_WR1.0.0ao.x86_64
net-tools-1.60-1_WRS1.0.0ao.1.x86_64
newt-0.52.2-1_WRS1.0.0ao.x86_64

nspr-4.8.2-1_WR1.0.0ao.x86_64
nss-3.12.4-14_WR1.0.0ao.x86_64
nss-tools-3.12.4-14_WR1.0.0ao.x86_64
ntp-4.2.4p0-1_WR1.0.0ao.x86_64
ntsysv-1.3.34-1_WR1.0.0ao.x86_64
openssh-5.0p1-1_WR1.0.0ao.1.x86_64
openssh-clients-5.0p1-1_WR1.0.0ao.1.x86_64
openssh-server-5.0p1-1_WR1.0.0ao.1.x86_64
openssl-0.9.8g-5_WR1.0.0ao.4.x86_64
openssl-perl-0.9.8g-5_WR1.0.0ao.4.x86_64
ospp-utils-1.0.0-1_WRS1.0.0ao.x86_64
pam-1.0.1-2_WR1.0.0ao.x86_64
pam_passwdqc-1.0.2-1.2.2_WR1.0.0ao.x86_64
paxctl-0.5-1_WRS1.0.0ao.x86_64
pciutils-2.2.4-3_WR1.0.0ao.x86_64
pciutils-data-2.2.4-3_WR1.0.0ao.x86_64
pcre-7.3-3_WR1.0.0ao.x86_64
perl-5.10.0-47_WR1.0.0ao.4.x86_64
perl-Archive-Extract-0.24-47_WR1.0.0ao.4.x86_64
perl-Archive-Tar-1.38-47_WR1.0.0ao.4.x86_64
perl-Compress-Raw-Zlib-2.008-47_WR1.0.0ao.4.x86_64
perl-Compress-Zlib-2.008-47_WR1.0.0ao.4.x86_64
perl-core-5.10.0-47_WR1.0.0ao.4.x86_64
perl-CPAN-1.9205-47_WR1.0.0ao.4.x86_64
perl-CPANPLUS-0.84-47_WR1.0.0ao.4.x86_64
perl-devel-5.10.0-47_WR1.0.0ao.4.x86_64
perl-Digest-SHA-5.45-47_WR1.0.0ao.4.x86_64
perl-ExtUtils-CBuilder-0.21-47_WR1.0.0ao.4.x86_64
perl-ExtUtils-Embed-1.27-47_WR1.0.0ao.4.x86_64
perl-ExtUtils-MakeMaker-6.36-47_WR1.0.0ao.4.x86_64
perl-ExtUtils-ParseXS-2.18-47_WR1.0.0ao.4.x86_64

perl-File-Fetch-0.14-47_WR1.0.0ao.4.x86_64
perl-IO-Compress-Base-2.008-47_WR1.0.0ao.4.x86_64
perl-IO-Compress-Zlib-2.008-47_WR1.0.0ao.4.x86_64
perl-IO-Zlib-1.07-47_WR1.0.0ao.4.x86_64
perl-IPC-Cmd-0.40-47_WR1.0.0ao.4.x86_64
perl-libs-5.10.0-47_WR1.0.0ao.4.x86_64
perl-Locale-Maketext-Simple-0.18-47_WR1.0.0ao.4.x86_64
perl-Log-Message-0.01-47_WR1.0.0ao.4.x86_64
perl-Log-Message-Simple-0.04-47_WR1.0.0ao.4.x86_64
perl-Module-Build-0.2808-47_WR1.0.0ao.4.x86_64
perl-Module-CoreList-2.14-47_WR1.0.0ao.4.x86_64
perl-Module-Load-0.12-47_WR1.0.0ao.4.x86_64
perl-Module-Load-Conditional-0.24-47_WR1.0.0ao.4.x86_64
perl-Module-Loaded-0.01-47_WR1.0.0ao.4.x86_64
perl-Module-Pluggable-3.60-47_WR1.0.0ao.4.x86_64
perl-Object-Accessor-0.32-47_WR1.0.0ao.4.x86_64
perl-Package-Constants-0.01-47_WR1.0.0ao.4.x86_64
perl-Params-Check-0.26-47_WR1.0.0ao.4.x86_64
perl-Pod-Escapes-1.04-47_WR1.0.0ao.4.x86_64
perl-Pod-Simple-3.05-47_WR1.0.0ao.4.x86_64
perl-suidperl-5.10.0-47_WR1.0.0ao.4.x86_64
perl-Term-UI-0.18-47_WR1.0.0ao.4.x86_64
perl-test-5.10.0-47_WR1.0.0ao.4.x86_64
perl-Test-Harness-3.12-47_WR1.0.0ao.4.x86_64
perl-Test-Simple-0.80-47_WR1.0.0ao.4.x86_64
perl-Time-Piece-1.12-47_WR1.0.0ao.4.x86_64
perl-version-0.74-47_WR1.0.0ao.4.x86_64
pils-2.1.3-2_WR1.0.0ao.x86_64
pmem-lib-3.0.1-1.x86_64
pmem-test-3.0.1-1.x86_64
pmem-tools-3.0.1-1.x86_64

policycoreutils-2.0.77-1_WRS1.0.0ao.x86_64
popt-1.14-1_WRS1.0.0ao.x86_64
portmap-4.0-65.2.2.1_WR1.0.0ao.x86_64
ppp-2.4.4-2_WR1.0.0ao.x86_64
prelink-150-1_WR4.3a_274.0.0.0.0.x86_64
procps-3.2.7-20_WR1.0.0ao.x86_64
psmisc-22.5-2_WR1.0.0ao.x86_64
pth-2.0.7-7_WR1.0.0ao.1.x86_64
pth-devel-2.0.7-7_WR1.0.0ao.1.x86_64
python-2.5.1-25_WR1.0.0ao.x86_64
python-elementtree-1.2.6-1_WR1.0.0ao.1.x86_64
python-libs-2.5.1-25_WR1.0.0ao.x86_64
quagga-0.99.10-2_WR1.0.0ao.x86_64
quagga-contrib-0.99.10-2_WR1.0.0ao.x86_64
quota-3.15-6_WR1.0.0ao.x86_64
rdist-6.1.5-45_WR1.0.0ao.x86_64
readline-5.2-1_WRS1.0.0ao.x86_64
refpolicy-2.20091117-1_WRS1.0.0ao.x86_64
refpolicy-strict-2.20091117-1_WRS1.0.0ao.x86_64
rng-tools-2-1_WRS1.0.0ao.x86_64
rpm-5.1.6-1_WRS1.0.0ao.x86_64
rsync-2.6.9-3.2_WR1.0.0ao.x86_64
samhain-2.5.5-1_WRS1.0.0ao.x86_64
screen-4.0.3-11_WR1.0.0ao.x86_64
scsidev-2.30-1_WRS1.0.0ao.x86_64
sed-4.1.5-9_WR1.0.0ao.x86_64
sepolgen-1.0.18-1_WRS1.0.0ao.x86_64
setools-3.3.5-0.x86_64
setools-console-3.3.5-0.x86_64
setools-libs-3.3.5-0.x86_64
setserial-2.17-20_WR1.0.0ao.x86_64

setup-2.8.9-1_WR1.0.0ao.noarch
shadow-utils-4.1.1-1_WR1.0.0ao.x86_64
slang-2.1.3-1_WRS1.0.0ao.x86_64
smartmontools-5.38-2_WR1.0.0ao.x86_64
sqlite-3.6.7-1_WRS1.0.0ao.x86_64
star-1.5.1-2.x86_64
stonith-2.1.3-2_WR1.0.0ao.x86_64
strace-4.5.15-1_WR1.0.0ao.x86_64
stunnel-4.31-1_WR1.0.0ao.x86_64
sudo-1.6.8p12-10_WR1.0.0ao.2.x86_64
sysfsutils-2.1.0-3_WR1.0.0ao.x86_64
syslog-ng-3.0.5-1_WR1.0.0ao.x86_64
sysstat-8.0.4-4_WR1.0.0ao.x86_64
SysVinit-2.86-14_WR1.0.0ao.x86_64
tar-1.17-3_WR1.0.0ao.1.x86_64
tcl-8.5.0-6_WR1.0.0ao.x86_64
tcpdump-3.9.5-3_WR1.0.0ao.x86_64
tcp_wrappers-7.6-44_WR1.0.0ao.x86_64
tcp_wrappers-libs-7.6-44_WR1.0.0ao.x86_64
timezone-2010j-1_WRS1.0.0ao.x86_64
tipc-utils-1.1.8-1_WR1.0.0ao.x86_64
traceroute-2.0.10-1_WR1.0.0ao.x86_64
udev-120-5.20080421git_WR1.0.0ao.x86_64
unionfs-1.1.5-1_WRS1.0.0ao.x86_64
ustr-1.0.4-6_WR1.0.0ao.1.x86_64
util-linux-ng-2.13.1-6_WR1.0.0ao.x86_64
vim-common-7.1.291-1_WR1.0.0ao.x86_64
vim-enhanced-7.1.291-1_WR1.0.0ao.x86_64
vim-minimal-7.1.291-1_WR1.0.0ao.x86_64
vixie-cron-4.1-69_WR1.0.0ao.x86_64
vlan-1.9-1_WRS1.0.0ao.x86_64

vlock-2.2.2-1_WR1.0.0ao.x86_64
watchdog-5.2.5-1.rf_WR1.0.0ao.x86_64
wcf-1.0-1_WRS1.0.0ao.x86_64
wdbagent-ptrace-3.2_58-1_WRS1.0.0ao.x86_64
wget-1.10.2-1_WRS1.0.0ao.x86_64
which-2.18-1_WRS1.0.0ao.x86_64
wrproxy-1.2-1_WRS1.0.0ao.x86_64
xerces-2.8.0-1_WRS1.0.0ao.x86_64
xinetd-2.3.14-3.1_WR1.0.0ao.x86_64
zlib-1.2.3-1.2.3_WR1.0.0ao.x86_64

1.5.2.4 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 UIDs, 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.

1.5.2.5 Security Policy Model

The security policy for Wind River Linux is defined by the security functional requirements in chapter 6. The following is a list of the subjects and objects participating in the policy.

Subjects:

- Processes acting on behalf of a human user or technical entity.

Named objects:

- File system objects in the following allowed file systems:
  - Ext3 - standard file system for general data
  - iso9660 - ISO9660 file system for CD-ROM and DVD
  - tmpfs - the temporary file system backed by RAM
  - rootfs - the virtual root file system used temporarily during system boot
  - procfs - process file system holding information about processes, general statistical data and tunable kernel parameters

  - sysfs - system-related file system covering general information about resources maintained by the kernel including several tunable parameters for these resources
  - devpts - pseudoterminal file system for allocating virtual TTYs on demand
  - binfmt_misc - configuration interface allowing the assignment of executable file formats with user space applications
  - securityfs - interface for loadable security modules (LSM) to provide tunables and configuration interfaces to user space
  - selinuxfs - interface for allowing user space components to interact with the SELinux module inside the kernel, including managing the SELinux policy.

  Please note that the TOE supports a number of additional virtual (i.e. without backing of persistent storage) file systems which are only accessible to the TSF - they are not or cannot be mounted. All above mentioned virtual file systems implement access decisions based DAC attributes inferred from the underlying process' DAC attributes. Additional restrictions may apply for specific objects in this file system.

- Inter Process Communication (IPC) objects:
  - Semaphores
  - Shared memory
  - Message queues
  - Named pipes
  - UNIX domain socket special files
  - Signals
- at job queue maintained for the root user
- cron job queues maintained for each user

TSF data:

- Subject meta data - all data used for subjects except data which is not interpreted by the TSF and does not implement parts of the TSF (this data is called user data)
- Named object meta data - all data used for the respective objects except data which is not interpreted by the TSF and does not implement parts of the TSF (this data is called user data)
- User accounts, including the security attributes defined by FIA_ATD.1
- Audit records

User data:

- Non-TSF executable code used to drive the behavior of subjects
- Data not interpreted by TSF and stored or transmitted using named objects

2 CC Conformance Claim

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

This ST claims conformance to the following Protection Profiles:

- [niap-ospp] : US Government Protection Profile for General-Purpose Operating Systems in a Networked environment. Version 1.0 as of 2010-08-30; demonstrable conformance.

Common Criteria [CC] version 3.1 revision 3 has been taken as the basis for this conformance claim.

2.1 Protection Profile tailoring and additions

This Security Target adds assumptions and objectives beyond those defined by the protection profile. Those assumptions and objectives cover additional functions or environmental constraints that are unrelated to the requirements in the protection profile.

The following list enumerates each SFR from the PP this ST claims compliance with which is modified in this ST beyond the operations allowed for this SFR. Any SPD component, objective and any other SFR is taken from the PP according to the rules for strict compliance.

FAU_GEN.1

The PP specifies in the table of audited events given with FAU_GEN.1.1 audit requirements which comply with either the basic or minimum level of audit defined by the CC for the respective SFR. In some instances, the table specifies the same requirements as the CC adopted to operating system terminology (such as FIA_AFL.1). In some cases, the PP does not make any auditing requirements even though minimal or basic level of auditing would require some (e.g. FIA_SOS.1). Therefore, the claim of this ST of basic level of auditing is consistent with the PP.

The following events to be audited are claimed by the PP which are beyond the basic level of auditing. The ST complies with the requirements as follows:

- FAU_GEN.1.1 "Uses of special permissions that circumvent the access control policies" is covered as these special permissions are modeled as part of FDP_ACF.1(1) and are therefore covered by the basic level of audit claim.
- FDP_ACF.1 "use of privilege to bypass the access control mechanism" is covered by the ST because the privileges that allow users to bypass the basic access control rule set are modeled as part of the access control rule set in FDP_ACF.1.3 as well as FDP_ACF.1.4. Therefore, the basic level of audit claimed for FDP_ACF.1 covers the auditing of these privileges.
- FTA_MCS.1 "Setting the limit on the number of multiple concurrent sessions by an authorized administrator." is covered as FMT_MTD.1(12) is specified for the management aspect of this SFR. This FMT_MTD.1 iteration again is subject to the basic level of auditing requirement.

FAU_GEN.1.2(b)

FAU_GEN.1.2(b) is extended by the list of audit information specified in the table 5.2 in the PP. The following audit information specified in the table 5.2 of the PP have not been added to FAU_GEN.1.2(b):

- FAU_STG.3: the message to be sent to the administrator is static which implies that it does not need to be audited. In addition, the TOE supports the feature of calling external applications (such as to trigger pager calls, sending emails or other operations) for which the TSF inherently does not possess any information about the submitted message.
- FMT_MSA.3: The specification of the initial default attributes is considered to be misleading as the user can specify alternative default values for his session at any time. Therefore, the TOE allows the auditing of the setting of the user's default values (e.g. umask system call, system call for setting the default ACL on directories) to allow administrators a clear picture of the state of the system.
- FMT_SMR.1: The role is implicitly assigned to users by considering the groups a user is assigned to. As the assignment of the user with a group is audited, the role assignment is implicitly covered.
- FPT_ITT.3: As this SFR is trivially met by the TOE, no audit record needs to be generated.

FAU_SAR.1.2

The PP requires that a tool is provided to interpret the audit events. As the TOE records the audit data as ASCII data, any tool would suffice. Hence, the revision of this SFR to CC Part 2 is appropriate.

FAU_SAR.3

The specification of this SFR has been extended to allow more operations during review as well as applying these operations to more audit entry attributes.

FAU_SEL.1

The specification of this SFR has been extended to allow more selection criteria that can be applied during the generation of audit records.

The requirement for selecting the host identity during audit generation has been removed as the TOE is not distributed, thus trivially satisfying this requirement. Note that for remote connections for, say, accessing the login mechanism, the TOE can record the IP address of the remote host.

FDP_ACF.1

The meta-rule set defined in the protection profile is refined with the rule set enforced by the TOE. An iteration of FDP_ACF.1 is provided specifying the access control rule for each named object type present in the TOE. Since the relevant application note in the protection profile states that the SFR is intended to be refined with a more restricted and more fine grained access control rules, the reversion of this SFR to CC Part 2 is appropriate.

FIA_SOS.1

The SFR has been rephrased to provide a probability argument for the quality of the password. The ST author considers the given probability to be in line with or even more restrictive than the PP requirements. The probability, however, includes the consideration of the duration for which a credential is valid. Password quality requirements of FIA_SOS.1 as specified in the PP allows the use of trivial passwords due to the specification of "any combination", allowing, say, 16 times the same character. To avoid trivial passwords but still allowing the administrator the most degree of freedom, the SFR in this ST specifies the probability of a successful password guessing attempt. The Evaluated Configuration Guide provides guidance for the minimum specification of the password quality control system to meet these requirements. Note, the guidance will include references for maintaining a password history as required by FIA_SOS.1.1(b) of the PP.

Considering the password space defined with the requirements in the PP, one cannot assume an equal distribution of the passwords. In fact, passwords tend to be heavily skewed based on the natural language used by the users. Without requiring a certain password quality metric in addition to a simple password length (as evident in the PP), trivial passwords are very likely to be used. In real life people are lazy. Who can memorize 16 char passwords which are of good quality if they are not forced to a certain pattern?

Worst-case scenario calculations show that much more stringent password quality rules (requiring at minimum 8 char for a password, at least one char out of each of the 4 character sets consisting of small alpha chars, capital alpha chars, numbers, special chars, prevention of keyboard patterns) barely exceed the probability of 2^-20.

FIA_USB.1

This SFR has been derived from CC Part 2. CC Part 2 requires the specification of rules for setting the security attributes of the subject. The PP, however, specifies the setting of the security attributes in databases which in turn are used on subjects eventually. The statements the PP makes for FIA_USB.1.2 and FIA_USB.1.3 are already specified in FMT_MTD.1(8), and FMT_REV.1(1). Therefore, this ST rephrases FIA_USB.1.2 and FIA_USB.1.3 to match the intention of CC Part 2.

FMT_MOF.1(2)

This SFR is completely removed as FMT_MTD.1(8) fully cover this SFR already.

FMT_MSA.4(1)

FMT_MSA.4(1) is added to clearly define the setting of the default DAC security attributes.

FMT_MTD.1(1)

This generic SFR has been replaced with a number of iterations of FMT_MTD.1 which specifically address the management aspect of individual SFRs to track that all such management aspects are addressed.

FMT_MTD.1(2)

This SFR has been extended to cover all aspects of the audit storage management.

FMT_MTD.1(3), FMT_MTD.1(4), FMT_MTD.1(5), FMT_MTD.1(6)

These SFRs have been collapsed into FMT_MTD.1(8) as the initialization (FMT_MTD.1(3)) and modification (FMT_MTD.1(4)) of user security attributes are defined in FMT_MTD.1(8). In addition, the authentication data (FMT_MTD.1(5)) belongs to the user security attributes and is therefore covered in FMT_MTD.1(8) as well. Also, the user is allowed read access to all user security attributes except authentication data (FMT_MTD.1(6)) as required by FMT_MTD.1(8).

FMT_SMF.1

This SFR has been updated to enumerate the general management capability of the TOE.

3 Security Problem Definition

3.1 Threat Environment

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** having an interest in manipulating the data model can be categorized as either:

- Unauthorized users of the TOE, i.e. human users or technical entities who have not been granted the right to access the system; or
- Authorized users of the TOE, i.e. human users or technical entities 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 enhanced-basic attack potential.

3.1.1 Threats countered by the TOE

T.ADMIN_ERROR

An administrator may incorrectly install or configure the TOE resulting in ineffective security mechanisms.

T.ADMIN_ROGUE

An authorized administrator’s intentions may become malicious resulting in user or TSF data being compromised.

T.AUDIT_COMPROMISE

A malicious user or process may view audit records, cause audit records to be lost or modified, or prevent future audit records from being recorded, thus masking a user’s action.

T.CRYPTO_COMPROMISE

A malicious user or process may cause key, data or executable code associated with the cryptographic functionality to be inappropriately accessed (viewed, modified, or deleted), thus compromising the cryptographic mechanisms and the data protected by those mechanisms.

T.MASQUERADE

A malicious user, process, or external IT entity may masquerade as an authorized entity in order to gain unauthorized access to data or TOE resources.

T.OPERATIONAL_ERRORS

While the TOE is operational, changes to the TOE may cause it to enter a configuration that is not able to enforce the security policies of the TOE.

T.RESIDUAL_DATA

A user or process may gain unauthorized access to data through reallocation of TOE resources from one user or process to another.

T.RESOURCE_EXHAUSTION

A malicious process or user may block others from system resources (i.e., persistent storage) via a resource exhaustion denial of service attack.

T.TSF_COMPROMISE

A malicious user or process may cause TSF data or executable code to be inappropriately accessed (viewed, modified or deleted).

T.UNATTENDED_SESSION

A user may gain unauthorized access to an unattended session.

T.UNAUTHORIZED_ACCESS

A user may gain unauthorized access (view, modify, delete) to user data.

T.UNIDENTIFIED_ACTIONS

The administrator may fail to notice potential security violations, thus preventing the administrator from taking action against a possible security violation.

T.UNKNOWN_STATE

When the TOE is initially started or restarted after a failure, the security state of the TOE may be unknown.

3.2 Assumptions

3.2.1 Environment of use of the TOE

3.2.1.1 Physical

A.PHYSICAL

It is assumed that the operational environment provides the TOE with appropriate physical security, commensurate with the value of the IT assets protected by the TOE.

3.2.1.2 Procedural

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., tape drives, disk drives) attached to the TOE, and marking a sensitivity label on all output generated.

3.2.1.3 Connectivity

A.CONNECT

All connections to peripheral devices and all network connections are protected against eavesdropping.

3.3 Organizational Security Policies

P.ACCESS_BANNER

The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the TOE.

P.ACCOUNTABILITY

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

P.AUTHORIZATION

The TOE shall limit the extent of each user's abilities in accordance with the TSP.

P.AUTHORIZED_USERS

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

P.CLASSIFICATION

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.

P.CRYPTOGRAPHY

The TOE shall use NIST FIPS validated cryptography as a baseline for key management (i.e., generation and destruction) and for cryptographic operations (i.e., encryption, decryption, signature, hashing, and random number generation services).

P.I_AND_A

All users must be identified and authenticated prior to accessing any controlled resources with the exception of public objects.

P.NEED_TO_KNOW

The TOE must limit the access to data in protected resources to those authorized users who have a need to know that data.

P.ROLES

The TOE shall provide multiple administrative roles for secure administration of the TOE. These roles shall be separate and distinct from each other.

P.TRACE

The TOE shall provide the ability to review the actions of individual users.

P.TRUSTED_RECOVERY

Procedures and/or mechanisms shall be provided to assure that, after a TOE failure or other discontinuity, recovery without a protection compromise is obtained.

4 Security Objectives

4.1 Objectives for the TOE

O.ACCESS

The TOE will ensure that users gain only authorized access to it and to resources that it controls.

O.ACCESS_HISTORY

The TOE will display information (to authorized users) related to previous attempts to establish a session.

O.ADMIN_ROLE

The TOE will provide administrator roles to isolate administrative actions.

O.AUDIT_GENERATION

The TOE will provide the capability to detect and create records of security relevant events associated with users.

O.AUDIT_PROTECTION

The TOE will provide the capability to protect audit information.

O.AUDIT_REVIEW

The TOE will provide the capability to selectively view audit information and alert the administrator of identified potential security violations.

O.CORRECT_TSF_OPERATION

The TOE will ensure the correct operation of the TSF by performing known-answer-tests with cryptographic mechanisms as well as verifying the integrity of the TSF executable code and TSF data throughout their lifetime.

O.CRYPTOGRAPHIC_SERVICES

The TOE will make encryption services available to authorized users and/or user applications.

O.DISCRETIONARY_ACCESS

The TOE will control access to resources based upon the identity of users and groups of users.

O.DISCRETIONARY_USER_CONTROL

The TOE will allow authorized users to specify which resources may be accessed by which users and groups of users.

O.DISPLAY_BANNER

The TOE will display an advisory warning regarding use of the TOE.

O.MANAGE

The TOE will provide all the functions and facilities necessary to support the authorized administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.

O.MANDATORY_ACCESS

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.PROTECT

The TOE will provide mechanisms to protect user data and resources.

O.RECOVERY

Procedures and/or mechanisms will be provided to assure that recovery is obtained without a protection compromise, such as from system failure or discontinuity.

O.RESIDUAL_INFORMATION

The TOE will ensure that any data contained in a protected resource is not available when the resource is reallocated.

O.RESOURCE_SHARING

The TOE shall provide mechanisms that mitigate user attempts to exhaust TOE resources (e.g., persistent storage).

O.REFERENCE_MONITOR

The TOE will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure.

O.TSF_CRYPTOGRAPHIC_INTEGRITY

The TOE will provide cryptographic integrity mechanisms for TSF data while in transit to remote parts of the TOE.

O.USER_AUTHENTICATION

The TOE will verify the claimed identity of users.

O.USER_IDENTIFICATION

The TOE will uniquely identify users.

4.2 Objectives for the Operational Environment

OE.PHYSICAL

Physical security will be provided for the TOE by the operational environment, commensurate with the value of the IT assets protected by the TOE.

OE.LABELING

Those responsible for the TOE must establish and implement procedures to ensure that information is protected in an appropriate manner, supporting the mandatory access control policy. MAC labeling of subjects and objects shall always be set up correctly.

4.3 Security Objectives Rationale

4.3.1 Security objectives coverage

The following table provides a mapping of TOE objectives to threats and policies, showing that each objective counters or enforces at least one threat or policy, respectively.

Objective	Threats / OSPs
O.ACCESS	T.UNAUTHORIZED_ACCESS P.AUTHORIZATION P.AUTHORIZED_USERS P.NEED_TO_KNOW
O.ACCESS_HISTORY	T.UNAUTHORIZED_ACCESS
O.ADMIN_ROLE	T.ADMIN_ROGUE P.ROLES
O.AUDIT_GENERATION	T.AUDIT_COMPROMISE P.ACCOUNTABILITY
O.AUDIT_PROTECTION	T.AUDIT_COMPROMISE
O.AUDIT_REVIEW	T.UNIDENTIFIED_ACTIONS P.ACCOUNTABILITY P.TRACE
O.CORRECT_TSF_OPERATION	T.OPERATIONAL_ERRORS
O.CRYPTOGRAPHIC_SERVICES	P.CRYPTOGRAPHY
O.DISCRETIONARY_ACCESS	P.NEED_TO_KNOW
O.DISCRETIONARY_USER_CONTROL	P.NEED_TO_KNOW
O.DISPLAY_BANNER	P.ACCESS_BANNER
O.MANAGE	T.ADMIN_ERROR
O.MANDATORY_ACCESS	P.CLASSIFICATION
O.PROTECT	T.UNATTENDED_SESSION T.UNAUTHORIZED_ACCESS P.AUTHORIZATION P.NEED_TO_KNOW
O.RECOVERY	T.UNKNOWN_STATE P.TRUSTED_RECOVERY
O.RESIDUAL_INFORMATION	T.RESIDUAL_DATA
O.RESOURCE_SHARING	T.RESOURCE_EXHAUSTION
O.REFERENCE_MONITOR	T.AUDIT_COMPROMISE T.CRYPTO_COMPROMISE T.TSF_COMPROMISE

Objective	Threats / OSPs
O.TSF_CRYPTOGRAPHIC_INTEGRITY	
O.USER_AUTHENTICATION	T.MASQUERADE P.I_AND_A
O.USER_IDENTIFICATION	T.MASQUERADE P.ACCOUNTABILITY P.AUTHORIZATION P.I_AND_A

Table 2: Mapping of security objectives to threats and policies

The following table provides a mapping of the objectives for the Operational Environment to assumptions, threats and policies, showing that each objective holds, counters or enforces at least one assumption, threat or policy, respectively.

Objective	Assumptions / Threats / OSPs
OE.PHYSICAL	A.PHYSICAL A.CONNECT T.AUDIT_COMPROMISE T.CRYPTO_COMPROMISE T.TSF_COMPROMISE T.UNAUTHORIZED_ACCESS
OE.LABELING	A.CLEARANCE A.SENSITIVITY

Table 3: Mapping of security objectives for the Operational Environment to assumptions, threats and policies

4.3.2 Security objectives sufficiency

The following rationale provides justification that the security objectives are suitable to counter each individual threat and that each security objective tracing back to a threat, when achieved, actually contributes to the removal, diminishing or mitigation of that threat:

Threat	Rationale for security objectives
T.ADMIN_ERROR	O.MANAGE contributes to mitigating this threat by providing the security mechanisms (e.g., tools for reviewing audit data) for administrators to perform TOE administration effectively, and to quickly alert the administrator of ineffective security policies on the TOE.
T.ADMIN_ROGUE	It is important to limit the functionality of administrative roles. If the intentions of an individual in an administrative role become malicious, O.ADMIN_ROLE mitigates this threat by isolating the administrative actions within that role and limiting the functions available to that

Threat	Rationale for security objectives
	individual. This objective presumes that separate individuals will be assigned separate distinct roles with no overlap of allowed operations between the roles.
T.AUDIT_COMPROMISE	O.AUDIT_GENERATION provides the capability to detect and create records of security relevant events. Audit records identify the user responsible for the event and are an important form of evidence that can be used to track an attacker's actions. Tampering with or destruction of audit data by physical means is addressed by OE.PHYSICAL, which provides physical security controls to the TOE environment. O.AUDIT_PROTECTION provides the capability to specifically protect audit information from external interference, tampering, or unauthorized disclosure. O.REFERENCE_MONITOR protects the TOE and its resources (including audit data) by ensuring that the security policies implemented by the TOE to protect the audit information are always invoked.
T.CRYPTO_COMPROMISE	The cryptography is afforded external protection from viewing, modification, or deletion by malicious users through physical security measures provided by the operational environment [OE.PHYSICAL]. Further, as part of the TOE's security functions (TSF), the cryptography is afforded internal protection from viewing, modification, or deletion by malicious processes and users through the domain isolation maintained by the TOE for its own execution [O.REFERENCE_MONITOR].
T.MASQUERADE	To address this threat, O.USER_IDENTIFICATION identifies the user as a legitimate user and O.USER_AUTHENTICATION authenticates this user preventing unauthorized users, processes, or external IT entities from masquerading as an authorized entity.
T.OPERATIONAL_ERRORS	The TOE must continue to operate correctly and enforce its security policies once it has been fielded. Some level of testing must be available to authorized users to ensure the TOE's security mechanisms continue to operate correctly once the TOE is fielded. O.CORRECT_TSF_OPERATION ensures that once the TOE is installed at a customer's location, the capability exists that the integrity of the TSF (hardware and software) can be demonstrated, and thus provides end users the confidence that the TOE's security policies continue to be enforced.
T.RESIDUAL_DATA	The sharing of hardware resources such as primary and secondary storage components between users introduces the potential for information flow in violation of the TOE security policy when hardware resources are deallocated from one user and allocated to another. In order to prevent such unintended consequences, the TOE prevents the compromise of the TOE security policy through mechanisms that ensure that residual information cannot be accessed after the resource has been reallocated (O.RESIDUAL_INFORMATION). The intent here is to prevent the unauthorized flow of information that would violate the TOE security policy. The intent is not to require explicit scrubbing or overwriting of data prior to reuse of the storage resource. Therefore, the

Threat	Rationale for security objectives
	presence of “residual” data in a storage resource is acceptable as long as it cannot be accessed by subsequent users such that a violation of the TOE security policy results.
T.RESOURCE_EXHAUSTION	The sharing of resources (i.e., persistent storage) between users introduces the potential for a malicious process or user to obstruct users from access to resources via a resource exhaustion denial-of-service attack. O.RESOURCE_SHARING mitigates this threat by requiring the TOE to provide controls to enforce maximum quotas for persistent storage.
T.TSF_COMPROMISE	The tampering with or destruction of TSF hardware, software, or configuration data via physical means is addressed by the physical security controls present in the TOE environment [OE.PHYSICAL]. O.REFERENCE_MONITOR addresses the threat of tampering with or destruction of TSF hardware, software, or configuration data by other (non-physical) means. It ensures that the TSF maintains a security domain for its own execution that protects it from interference and tampering by untrusted subjects and enforces the separation between the security domains of subjects within the TSC.
T.UNATTENDED_SESSION	When an authorized user leaves an active session unattended, an unauthorized user may gain access to the unattended session. O.PROTECT mitigates this threat by providing mechanisms to protect user data and resources from unauthorized access by ensuring that the TSF will lock an interactive session and make the visible contents unreadable after a specified time interval of session inactivity.
T.UNAUTHORIZED_ACCESS	Unauthorized users may physically access TOE resources. To mitigate this threat, OE.PHYSICAL restricts the physical access only to authorized personnel. Within the computing environment, O.ACCESS restricts all access controls to authorized users based on their user identity. At the same time, O.PROTECT enforces access rules by providing mechanisms to prevent the user data from unauthorized disclosure and modification. O.ACCESS_HISTORY helps users confirm their previously established session or may help detected possible unsuccessful attempts to their account by an unauthorized user.
T.UNIDENTIFIED_ACTIONS	The threat of an administrator failing to know about audit events may occur. To mitigate this threat, O.AUDIT_REVIEW provides the capability to selectively view audit information, and alert the administrator of identified potential security violations.
T.UNKNOWN_STATE	After a failure, the security condition of the TOE may be unknown. To mitigate this threat O.RECOVERY provides procedures and/or mechanisms to ensure that recovery without a protection compromise is obtained.

Table 4: Sufficiency of objectives countering threats

The following rationale provides justification that the security objectives for the environment are suitable to cover each individual assumption, that each security objective for the environment that traces back to an assumption about the environment of use of the TOE, when achieved, actually contributes to the environment achieving consistency with the assumption, and that if all security objectives for the environment that trace back to an assumption are achieved, the intended usage is supported:

Assumption	Rationale for security objectives
A.PHYSICAL	Physical security must be provided for the TOE by the operational environment to ensure the TOE is capable of addressing the threats to TOE assets [OE.PHYSICAL].
A.CLEARANCE	The assumption on the procedures for granting authorization for access to specific security levels is covered by OE.LABELING which requires that MAC protections are set up correctly.
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.LABELING which requires that MAC protections are set up correctly.
A.CONNECT	Physical security must be provided for the TOE by the operational environment as defined by OE.PHYSICAL to ensure the TOE is capable of addressing the threats to the networking aspect of the TOE. Note, the physical protection of the entire TOE is required by A.PHYSICAL 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.

Table 5: Sufficiency of objectives holding assumptions

The following rationale provides justification that the security objectives are suitable to cover each individual organizational security policy, that each security objective that traces back to an OSP, when achieved, actually contributes to the implementation of the OSP, and that if all security objectives that trace back to an OSP are achieved, the OSP is implemented:

OSP	Rationale for security objectives
P.ACCESS_BANNER	O.DISPLAY_BANNER satisfies this policy by ensuring that the TOE displays a banner that provides authorized users with an advisory warning about the unauthorized use of the TOE.
P.ACCOUNTABILITY	Enforcement of this policy requires that users be uniquely identified [O.USER_IDENTIFICATION] and that their security relevant actions be monitored and recorded [O.AUDIT_GENERATION]. The recorded audit information can be selectively reviewed in search of any potential security violations [O.AUDIT_REVIEW].
P.AUTHORIZATION	O.ACCESS supports this policy by requiring the TOE to uniquely identify authorized users [O.USER_IDENTIFICATION] prior to allowing any TOE access or any TOE mediated access on behalf of those users.

OSP	Rationale for security objectives
	Within the TOE, O.PROTECT provides mechanisms to prevent user data from unauthorized disclosure and modification.
P.AUTHORIZED_USERS	Within the set of all the users that may interact with the TOE, authorized users are those with access to the information within the TOE after being successfully identified and authenticated by the TOE. Access control policies are used to define the access permitted to the system and its resources. These policies are supported by the implementation of authorized user attributes that identify the user-allowed accesses to TOE information. O.ACCESS supports this policy by ensuring that users only gain authorized access to TOE information and its resources by checking user attributes before system use.
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.
P.CRYPTOGRAPHY	By building upon NIST FIPS-validated, cryptography, the TOE not only provides, but also augments the cryptographic support offered solely by baseline NIST FIPS-validated cryptography. The TOE cryptography supports key management (i.e., generation and destruction of keys) and cryptographic operations (i.e., encryption, decryption, signature, hashing, and random number generation). O.CRYPTOGRAPHIC_SERVICES provides these cryptographic services to TOE authorized users and/or user applications.
P.I_AND_A	In support of the policy to identify and authenticate a user before access is granted to any controlled resources, O.USER_IDENTIFICATION and O.USER_AUTHENTICATION will uniquely identify and authenticate the claimed authorized users.
P.NEED_TO_KNOW	The need-to-know policy is satisfied by the discretionary access control rules. O.DISCRETIONARY_ACCESS protects resources based on the identity of authorized users where the access to objects is directed by owners of the object [O.DISCRETIONARY_USER_CONTROL]. O.PROTECT enforces these policy rules by providing the mechanisms to protect the user data from disclosure and modifications and lastly, O.ACCESS ensures that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.
P.ROLES	To appropriately administer the system, O.ADMIN_ROLE requires the system to provide multiple administrator roles to isolate actions performed by these different roles. To completely satisfy this policy, separate roles must be assigned separate individuals.
P.TRACE	A common organizational security policy is to maintain records allowing for individuals to be held responsible for the actions that they take with respect to organizational assets. Information can be one of the most valuable assets that an organization possesses. To satisfy this policy, O.AUDIT_REVIEW provides suitable mechanisms to accurately and

OSP	Rationale for security objectives
	selectively review those records by authorized personnel to provide accountability at the individual user level to determine any potential security violation.
P.TRUSTED_RECOVERY	After a failure or other discontinuity, the security condition of the TOE may be unknown. O.RECOVERY provides procedures and/or mechanisms to ensure that recovery to a known secure state is obtained without a protection compromise.

Table 6: Sufficiency of objectives enforcing Organizational Security Policies

5 Extended Components Definition

The Security Target includes the extended components defined by the PP. These extended components are not defined in this section.

In addition, the Security Target defines the extended component of FDP_RIP.3 as part of the FDP_RIP family in CC Part 2 for usage within this ST.

5.1 Class FDP: User data protection

5.1.1 (RIP)

Component levelling

FDP_RIP.3 is not hierarchical to any other component within the FDP_RIP family.

Management: FDP_RIP.3

The following actions could be considered for the management functions in FMT:

a) See management description specified for FDP_RIP.2 in CC Part 2.

Audit: FDP_RIP.3

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: See audit requirement specified for FDP_RIP.2 in CC Part 2.
b) Basic: See audit requirement specified for FDP_RIP.2 in CC Part 2.
c) Detailed: See audit requirement specified for FDP_RIP.2 in CC Part 2.

5.1.1.1 FDP_RIP.3 - Full residual information protection of resources

Hierarchical to:	No other components.
Dependencies:	No dependencies.

FDP_RIP.3 The TSF shall ensure that any previous information content of a resource is made unavailable upon the [selection: **allocation of the resource to, de-allocation of the resource from**] all subjects or users.

6 Security Requirements

6.1 TOE Security Functional Requirements

The following table shows the Security functional requirements for the TOE, and the operations performed on the components according to CC part 2: iteration (Iter.), refinement (Ref.), assignment (Ass.) and selection (Sel.).

Security functional class	Security functional requirement	Security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
FAU - Security audit	FAU_GEN.1 Audit data generation		CC Part 2	No	No	Yes	Yes
	FAU_GEN.2 User identity association		NIAP-OSPP	No	No	No	No
	FAU_SAR.1 Audit Review		CC Part 2	No	No	Yes	No
	FAU_SAR.2 Restricted Audit Review		NIAP-OSPP	No	No	No	No
	FAU_SAR.3 Selectable Audit Review		CC Part 2	No	No	Yes	No
	FAU_SEL.1 Selective Audit		NIAP-OSPP	No	No	Yes	No
	FAU_STG.1 Protected Audit Trail Storage		NIAP-OSPP	No	No	No	No
	FAU_STG.3 Action in case of possible audit data loss		NIAP-OSPP	No	No	No	No
	FAU_STG.4 Prevention of audit data loss		CC Part 2	No	Yes	Yes	Yes
FCS - Cryptographic support	FCS_BCM_EXT.1 Baseline Cryptographic Module		NIAP-OSPP	No	No	No	No
	FCS_CKM.1(1) Cryptographic Key Generation (for symmetric keys)	FCS_CKM.1	NIAP-OSPP	Yes	No	Yes	No
	FCS_CKM.1(2) Cryptographic Key Generation (for asymmetric keys)	FCS_CKM.1	NIAP-OSPP	Yes	No	Yes	Yes
	FCS_CKM.4 Cryptographic Key Destruction		NIAP-OSPP	No	No	No	No
	FCS_COA_EXT.1 Cryptographic Operations Availability		NIAP-OSPP	No	No	Yes	No
	FCS_COP.1(1) Cryptographic Operation (for data encryption/decryption)	FCS_COP.1	NIAP-OSPP	Yes	No	Yes	Yes

Security functional class	Security functional requirement	Security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
	FCS_COP.1(2) Cryptographic Operation (for cryptographic signature)	FCS_COP.1	NIAP-OSPP	Yes	Yes	No	Yes
	FCS_COP.1(3) Cryptographic Operation (for cryptographic hashing)	FCS_COP.1	NIAP-OSPP	Yes	Yes	No	Yes
	FCS_RBG_EXT.1 Random Num ber Generation		NIAP-OSPP	No	Yes	No	Yes
FDP - User data protection	FDP_ACC.2 Complete Access Control		NIAP-OSPP	No	Yes	No	No
	FDP_ACF.1(1) Security Attribute Based Access Control (File Sys tem Objects)	FDP_ACF.1	CC Part 2	Yes	Yes	Yes	No
	FDP_ACF.1(2) Security Attribute Based Access Control (IPC Ob jects)	FDP_ACF.1	CC Part 2	Yes	Yes	Yes	No
	FDP_ACF.1(4) Security Attribute Based Access Control (at and cron job queues)	FDP_ACF.1	CC Part 2	Yes	Yes	Yes	No
	FDP_ETC.1 Export of unlabeled user data		CC Part 2	No	Yes	Yes	No
	FDP_ETC.2 Export of labeled user data		CC Part 2	No	Yes	Yes	No
	FDP_IFC.2 Mandatory Access Control Policy		CC Part 2	No	Yes	Yes	No
	FDP_IFF.2 Mandatory Access Control Functions		CC Part 2	No	Yes	Yes	No
	FDP_ITC.1 Import of unlabeled user data		CC Part 2	No	Yes	Yes	No
	FDP_ITC.2 Import of labeled user data		CC Part 2	No	Yes	Yes	No
	FDP_RIP.2 Full Residual Informa tion Protection		CC Part 2	No	No	No	No
	FDP_RIP.3 Full Residual Informa tion Protection of resources		ECD	No	No	No	Yes
FIA - Identification and authentication	FIA_AFL_EXT.1 Authentication Failures		NIAP-OSPP	No	No	No	No

Security functional class	Security functional requirement	Security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
	FIA_ATD.1 User attribute definition		NIAP-OSPP	No	Yes	Yes	No
	FIA_SOS.1 Verification of Secrets		CC Part 2	No	No	Yes	No
	FIA_UAU.1 Timing of authentication		NIAP-OSPP	No	No	No	No
	FIA_UAU.6 Re-authenticating		NIAP-OSPP	No	No	No	No
	FIA_UAU.7 Protected authentication feedback		NIAP-OSPP	No	No	No	No
	FIA_UID.1 Timing of identification		NIAP-OSPP	No	No	No	No
	FIA_USB.1 User-subject binding		CC Part 2	No	No	Yes	No
FMT - Security management	FMT_MOF.1 Management of Functions in TSF		NIAP-OSPP	No	No	No	No
	FMT_MSA.1(1) Management of Security Attributes (for Discretionary and Mandatory Access Control)	FMT_MSA.1	NIAP-OSPP	Yes	Yes	No	No
	FMT_MSA.1(2) Management of Security Attributes (for Object Ownership)	FMT_MSA.1	NIAP-OSPP	Yes	Yes	No	No
	FMT_MSA.2 Secure Security Attributes		NIAP-OSPP	No	No	No	No
	FMT_MSA.3(1) Static attribute initialisation	FMT_MSA.3	NIAP-OSPP	Yes	No	No	No
	FMT_MSA.3(2) Static attribute initialisation	FMT_MSA.3	CC Part 2	Yes	No	Yes	Yes
	FMT_MSA.4(1) Security attribute value inheritance (DAC)	FMT_MSA.4	CC Part 2	Yes	Yes	Yes	No
	FMT_MSA.4(2) Security attribute value inheritance (MAC)	FMT_MSA.4	CC Part 2	Yes	Yes	Yes	No
	FMT_MTD.1(1) Management of TSF Data (Audited Events)	FMT_MTD.1	CC Part 2	Yes	No	Yes	Yes
	FMT_MTD.1(2) Management of TSF Data (Audit Storage)	FMT_MTD.1	CC Part 2	Yes	No	Yes	Yes
	FMT_MTD.1(3) Management of TSF Data (Audit Threshold)	FMT_MTD.1	CC Part 2	Yes	No	Yes	Yes

Security functional class	Security functional requirement	Security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
	FMT_MTD.1(4) Management of TSF Data (Audit Storage Failure)	FMT_MTD.1	CC Part 2	Yes	No	Yes	Yes
	FMT_MTD.1(5) Management of TSF Data (Authentication Failure Threshold)	FMT_MTD.1	CC Part 2	Yes	No	Yes	Yes
	FMT_MTD.1(6) Management of TSF Data (Authentication Failure Re-enabling)	FMT_MTD.1	CC Part 2	Yes	No	Yes	Yes
	FMT_MTD.1(7) Management of TSF Data (for critical cryptographic security parameters)	FMT_MTD.1	NIAP-OSPP	Yes	No	No	No
	FMT_MTD.1(8) Management of TSF Data (User Security Attributes)	FMT_MTD.1	CC Part 2	Yes	No	Yes	Yes
	FMT_MTD.1(9) Management of TSF Data (Password quality)	FMT_MTD.1	CC Part 2	Yes	No	Yes	Yes
	FMT_MTD.1(10) Management of TSF Data (Label mapping rules)	FMT_MTD.1	CC Part 2	Yes	No	Yes	Yes
	FMT_MTD.1(11) Management of TSF Data (File system quotas)	FMT_MTD.1	CC Part 2	Yes	No	Yes	Yes
	FMT_MTD.1(12) Management of TSF Data (Maximum concurrent sessions)	FMT_MTD.1	CC Part 2	Yes	No	Yes	Yes
	FMT_MTD.1(13) Management of TSF Data (Session Locking)	FMT_MTD.1	CC Part 2	Yes	No	Yes	Yes
	FMT_MTD.1(14) Management of TSF Data (TOE Banner)	FMT_MTD.1	CC Part 2	Yes	No	Yes	Yes
	FMT_REV.1(1) Revocation (User security attributes)	FMT_REV.1	NIAP-OSPP	Yes	No	No	No
	FMT_REV.1(2) Revocation (Access permissions)	FMT_REV.1	NIAP-OSPP	Yes	Yes	No	No
	FMT_SAE.1 Time-limited authorization		NIAP-OSPP	No	No	No	No
	FMT_SMF.1 Specification of Management Functions		CC Part 2	No	No	Yes	No
	FMT_SMR.1 Security Roles		NIAP-OSPP	No	No	Yes	No

Security functional class	Security functional requirement	Security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
FPT - Protection of the TSF	FPT_ITT.1 Basic Internal TSF Data Transfer Protection		NIAP-OSPP	No	No	Yes	No
	FPT_ITT.3 TSF Data Integrity Monitoring		NIAP-OSPP	No	No	Yes	No
	FPT_RCV.1 Manual Recovery		NIAP-OSPP	No	No	No	No
	FPT_STM.1 Reliable Time Stamps		NIAP-OSPP	No	No	No	No
	FPT_TRC_EXT.1 Internal TSF Data Consistency		NIAP-OSPP	No	No	No	No
	FPT_TDC.1 Inter-TSF basic TSF data consistency		CC Part 2	No	No	Yes	No
	FPT_TST_EXT.1 TSF Testing		NIAP-OSPP	No	No	No	No
FRU - Resource utilisation	FRU_RSA.1 Maximum Quotas		NIAP-OSPP	No	No	No	No
FTA - TOE access	FTA_MCS.1 Basic limitation on multiple concurrent sessions		NIAP-OSPP	No	No	No	No
	FTA_SSL.1 TSF-Initiated Session Locking		NIAP-OSPP	No	No	No	No
	FTA_SSL.2 User-Initiated Lock ing		NIAP-OSPP	No	No	No	No
	FTA_TAB.1 Default TOE access banners		NIAP-OSPP	No	No	No	No
	FTA_TAH.1 TOE Access History		NIAP-OSPP	No	No	No	No

Table 7: Security functional requirements for the TOE

6.1.1 Security audit (FAU)

6.1.1.1 Audit data generation (FAU_GEN.1)

FAU_GEN.1.1 The TSF shall be able to generate an audit record of the following auditable events:

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

b) Start-up and shutdown of the **basic** level of audit;

c) **all modifications to the set of events being audited;**

d) **all denied accesses to objects;**

e) **explicit modifications of access rights to objects covered by the access control policies;**

f) **(FCS_BCM_EXT.1) Failure of the cryptographic operation;**

g) **(FCS_RBG_EXT.1) Failure in the randomization process.**

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 (if applicable), and the outcome (success or failure) of the event; and

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

1. **User identity (if applicable);**
2. **(FAU_SAR.1) Name of object (audit log file);**
3. **(FCS_CKM.4) Identity of subject requesting or causing zeroization, identity of object or entity being cleared;**
4. **(FCS_COP.1(1)) Cryptographic mode of operation, name of object being encrypted/decrypted;**
5. **(FCS_COP.1(2)) Cryptographic mode of operation, name of object being signed/verified;**
6. **(FCS_COP.1(3) Cryptographic mode of operation, name of object being hashed;**
7. **(FDP_ACF.1 - all iterations) The name of the object being accessed;**
8. **(FIA_UAU.1) Origin of the attempt (e.g., terminal identifier, source IP address);**
9. **(FIA_UAU.6) Origin of the attempt (e.g., terminal identifier, source IP address);**
10. **(FIA_UID.1) Provided user identity, origin of the attempt (e.g., terminal identifier, source IP address);**
11. **(FMT_MOF.1(1)) The old and new values for audit events specified by this function;**
12. **(FMT_MSA.1(1)) The name of the object, the old and new values of the attributes;**
13. **(FMT_MSA.2) All offered and rejected values for a security attribute;**
14. **(FMT_MTD.1 - all iterations) The old and new values of the TSF data except authentication data or other sensitive data, as applicable;**
15. **(FMT_REV.1(1)) The security attributes that are attempting to be revoked;**
16. **(FMT_REV.1(2)) The security attributes that are attempting to be revoked, the object with which the security attributes are associated;**
17. **(FPT_RCV.1) Type of failure or service discontinuity;**
18. **(FPT_STM.1) The old and new values for the time;**
19. **(FPT_TST.1) For each test, the identification of the test and the results of that test;**
20. **(FRU_RSA.1) Object or other entity associated with failed allocation operation;**

21. **(FTA_MCS.1) The old and new values of the number of multiple concurrent sessions (for setting the session limit).**

6.1.1.2 User identity association (FAU_GEN.2)

FAU_GEN.2.1 For audit events resulting from actions of identified users, 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 UID”, 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 UID e. g. using the su command or executing a setuid or setgid program.*

6.1.1.3 Audit Review (FAU_SAR.1)

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

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

6.1.1.4 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 and MAC permissions ensure that only authorized administrators have access to the audit records.*

6.1.1.5 Selectable Audit Review (FAU_SAR.3)

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

a) **User identity (real, effective, filesystem),**
b) **Group identifier (real, effective, filesystem),**
c) **Event type,**
d) **Outcome (success/failure),**
e) **Login from a specific remote hostname,**
f) **Login UID,**
g) **Process ID,**
h) **Date and time of the audit event,**
i) **Object identity,**
j) **Subject sentitvity label;**
k) **Object sentitvity label.**

6.1.1.6 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) **Type of audit event;**
b) **Subject (process ID) or user identity;**
c) **Outcome (success or failure) of the audit event;**
d) **Named object identity;**
e) **Access types on a particular object;**
f) **System call number;**
g) **Subject sentitvity label;**
h) **Object sentitvity label.**

6.1.1.7 Protected Audit Trail Storage (FAU_STG.1)

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

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

6.1.1.8 Action in case of possible audit data loss (FAU_STG.3)

FAU_STG.3.1 The TSF shall notify an authorized administrator of the possible audit data loss if the audit trail exceeds an authorized administrator selectable, pre-defined limit.

Application Note: *The alarm generated by the TOE can be configured to be a syslog message or the execution of an administrator-specified application. This message or action of executing the application is generated when the audit trail capacity exceeds the limit defined in the auditd.conf file.*

6.1.1.9 Prevention of audit data loss (FAU_STG.4)

FAU_STG.4.1 The TSF shall ***be able to*** **prevent audited events, except those taken by the authorised** ~~**user with special rights**~~ ***administrator*** and **perform one of the following administrator-defined actions:**

a) **Stop all processes that attempt to generate an audit record;**
b) **Switch to single user mode;**
c) **Halt the system**

if the audit trail is full.

Application Note: *The TOE stops processes that want to generate or trigger an operation configured 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. 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.*

6.1.2 Cryptographic support (FCS)

6.1.2.1 Baseline Cryptographic Module (FCS_BCM_EXT.1)

FCS_BCM_EXT.1.1 All FIPS-approved cryptographic functions implemented by the TSF shall be implemented in a cryptomodule that is FIPS 140-2 validated, and perform the specified cryptographic functions in a FIPS-approved mode of operation. The FIPS 140-2 validation shall include an algorithm validation certificate for all FIPS-approved cryptographic functions.

6.1.2.2 Cryptographic Key Generation (for symmetric keys) (FCS_CKM.1(1))

FCS_CKM.1.1 The TSF shall generate cryptographic keys using a FIPS-Approved Random Number Generator as specified in FCS_RBG_EXT.1, and provide integrity protection to generated keys that leave the cryptomodule in accordance with NIST SP 800-57 “Recommendation for Key Management—Part 1: General,” paragraph 6.2.2.2a. in the following manner: **SHA-1, SHA-224, SHA-256, SHA384, SHA-512** .

Application Note: *The TOE utilizes the DRBG based deterministic random number generator which is implemented with the NSS library.*

Application Note: *The WCF provides the generic cryptographic functionality by providing a PKCS#11 interface. The PKCS#11 function of C_WrapKey provides the wrapping functionality. This function must be invoked with one parameter specifying the wrapping mechanism (CK_MECHANISM_PTR). The wrapped key can be protected by specifying symmetric or asymmetric ciphers for encryption as well as digest mechanism for hashing.*

6.1.2.3 Cryptographic Key Generation (for asymmetric keys) (FCS_CKM.1(2))

FCS_CKM.1.1 The TSF shall generate asymmetric cryptographic keys in accordance with domain parameter sizes **for ECDSA-based keys, 256 bits, 384 bits, 512 bits** that meet the following: FIPS 140-2.

6.1.2.4 Cryptographic Key Destruction (FCS_CKM.4)

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a cryptographic key zeroization method that meets the following: Key zeroization requirements of FIPS PUB 140-2, “Security Requirements for Cryptographic Modules”.

Application Note: *The NSS library stores the private key material or the critical security parameters (CSPs) in a database that is encrypted. The encryption key is unlocked with a user-supplied password. Zeroization implies that the NSS library makes the information to be zeroized inaccessible.*

6.1.2.5 Cryptographic Operations Availability (FCS_COA_EXT.1)

FCS_COA_EXT.1.1 The TSF shall provide the following cryptographic operations to applications:

a) Encryption/Decryption,

b) Cryptographic Signature (Digital Signature),

c) Hashing, and

d) **no other cryptographic operations.**

Application Note: *All cryptographic operations covered by security claims in this ST are implemented with the NSS library. To prevent eavesdropping of the operation of the NSS library, the NSS library is supplemented with a wrapper application that has to be invoked by the subject requesting services from NSS. This way, the calling subject cannot interfere with the operation or access the operational state of the NSS library.*

6.1.2.6 Cryptographic Operation (for data encryption/decryption) (FCS_COP.1(1))

FCS_COP.1.1 The TSF shall perform encryption and decryption using the FIPS-approved security function AES algorithm operating in **CBC, ECB mode** and cryptographic key size of **128 bits, 192 bits, 256 bits** that meets FIPS 140-2.

6.1.2.7 Cryptographic Operation (for cryptographic signature) (FCS_COP.1(2))

FCS_COP.1.1 The TSF shall perform cryptographic signature services using the FIPS-approved security function **Elliptic Curve Digital Signature Algorithm (ECDSA) ~~with a key size of 256 bits, 384 bits, 521 bits,~~ using only the NIST curve(s) P-256, P-384, P-521 as defined in FIPS PUB 186-3, “Digital Signature Standard”** that meets FIPS 140-2.

Application Note: *The refinement is due to the fact that ECDSA does not specify key sizes in bits/bytes for its operation but with a reference to the underlying elliptic curve.*

6.1.2.8 Cryptographic Operation (for cryptographic hashing) (FCS_COP.1(3))

FCS_COP.1.1 The TSF shall perform cryptographic hashing services in accordance with ~~SHA 256, SHA 384, SHA 512~~ *SHA2* and message digest sizes **256, 384, 512** bits that meet the following: FIPS 140-2.

Application Note: *The refinement prevents an ambiguous statement between the hash types and the message digest key lengths.*

6.1.2.9 Random Number Generation (FCS_RBG_EXT.1)

FCS_RBG_EXT.1.1 The TSF shall perform all random bit generation (RBG) services in accordance with **FIPS Pub 140-2 Annex C** implemented in a FIPS-validated cryptomodule operating in FIPS mode seeded by an entropy source that accumulates entropy from **a combination of hardware-based and software-based noise sources**.

Application Note: *The TOE implements the DRBG.*

Application Note: *The source for the seed and seed key is /dev/urandom which is the non-blocking random number generator provided by the Linux kernel i.e. this random number generator is a hybrid hardware-based and deterministic random number generator where the entropy is primarily gathered from hardware events. In case the estimated entropy of the gathered hardware events is considered to be insufficient, the deterministic randon number generator is used seeded with the hardware entropy pool. Once the hardware events provide again sufficient estimated entropy, the deterministic random number generator is deactivated again.*

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded with a minimum of ***440 bits*** of *statistically estimated* entropy at least equal to the greatest bit length of the keys that it will generate.

Application Note: *The refinement of using 440 bits is considered to be appropriate as this seed size is much larger than the sizes required by the PP implying that potentially more entropy is added to the deterministic RBG.*

Application Note: *The DRBG implementation fetches 880 bits from /dev/urandom where one half is considered to be the entropy and the second half the nonce.*

Application Note: *No RBG can determine the real entropy of the seed data. Therefore, the refinement for specifying that the entropy is statistically estimated is added. The statistical analysis is performed as part of the /dev/urandom implementation inside the Linux kernel. The statistical analysis is based on a worst-case scenario to provide statistically-assured a minimum entropy.*

6.1.3 User data protection (FDP)

6.1.3.1 Complete Access Control (FDP_ACC.2)

FDP_ACC.2.1 The TSF shall enforce the Discretionary Access Control policy on all subjects and all named objects and all operations among them.

Application Note: *The subjects and named objects covered in the TOE are defined with Security Policy Model .*

FDP_ACC.2.2 The TSF shall ensure that all operations between any subject controlled by the TSF and any *named* object controlled by the TSF are covered by an access control SFP.

6.1.3.2 Security Attribute Based Access Control (File System Objects) (FDP_ACF.1(1))

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

a) **Subject security attributes: file system UID, file system GID, supplemental GIDs;**

b) **Object security attributes: owning UID, owning GID;**

c) **Access control security attributes maintained for each file system object governing access to that object:**

   1. **ACL for specific UIDs (ACL_USER),**
   2. **ACL for specific GIDs (ACL_GROUP),**
   3. **Maximum ACL for the file system object (ACL_MASK),**
   4. **Permission bits for the owning UID (equals to ACL_USER_OBJ when using ACLs),**
   5. **Permission bits for the owning GID (equals to ACL_GROUP_OBJ when using ACLs),**
   6. **Permission bits for "world" (equals to ACL_OTHER when using ACLs),**

7. **The following permission bits: read, write, execute (for files), search (for directories),**
8. **The following access rights applicable to the file system object: SAVETXT (directories), immutable (files),**

d) **Access control security attributes maintained for each partition holding a file system: read-only;**

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

A 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 one of the following rules hold (the order of the rules is applicable on a first-match basis):

- **The subject's filesystem UID is identical with the owning UID of the object and the requested type of access is within the permission bits defined for the owning UID (permission bits) or by ACL_USER_OBJ (ACLs); or**
- **ACLs: The subject's filesystem UID is identical with the UID specified with ACL_USER of the object and the requested type of access is within the permission bits defined in ACL_USER; or**
- **The subject's filesystem GID or one of the subject's supplemental GIDs identical with the owning GID and the requested type of access is within the permission bits defined for the owning GID (permission bits), or by ACL_GROUP_OBJ when there is no ACL_MASK entry (ACLs), or by the ACL_MASK entry (ACLs); or**
- **ACLs: The subject's filesystem GID or one of the subject's supplemental GIDs is identical with the GID specified with ACL_GROUP of the object and the requested type of access is within the permission bits defined in ACL_GROUP; or**
- **The requested type of access is within the permission bits defined for "world" (permission bits) or by ACL_OTHER (ACLs).**

Application Note: *The permission bits and the ACLs are inherently consistent as the TOE assigns the permission bits to ACLs when ACLs are used. Without any ACLs specified for an object, the TOE only uses the permission bits. If at least one ACL is present or when the ACL management tools are applied for objects even without any ACL set, the permission bits are interpreted as outlined above: the ACL entry of ACL_USER_OBJ contains the owning UID permission bits, the ACL entry of ACL_GROUP_OBJ contains the owning GID permission bits, and the ACL entry of ACL_OTHER contains the permission bits for "world". The ACL entries of ACL_USER_OBJ, ACL_GROUP_OBJ and ACL_OTHER are only a different representation of the permission bits to users, they are not separate attributes in addition to permission bits. The explicit specification of ACL_USER_OBJ, ACL_GROUP_OBJ and ACL_OTHER in the rule set above in addition to the permission bits is only intended to aid the evaluator or reader in understanding the overall ruleset.*

Application Note: *Due to the fact that the permission bits are an inherent part of the ACLs, there is no precedence issue between permission bits and ACLs.*

FDP_ACF.1.3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: **all operations except execute operations are allowed for the subject with the file system UID of zero - the execute permission is granted if the file system object object is marked with at least one executable bit in its permission settings**.

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

a) **Any file system object in a file system that is mounted as read-only cannot be modified, created or removed,**

b) **Any file system object marked as immutable cannot be modified or removed,**

c) **Any file system object stored in a directory marked with the SAVETXT bit cannot be modified or removed by subjects whose file system UID is not equal to the owning UID of the file system object.**

6.1.3.3 Security Attribute Based Access Control (IPC Objects) (FDP_ACF.1(2))

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

a) **Subject security attributes: effective UID, effective GID, supplemental GIDs;**

b) **Object security attributes: owning UID, creator UID, owning GID;**

c) **Access control security attributes maintained for each IPC object except signals governing access to that object:**

   1. **Permission bits for the owning UID,**
   2. **Permission bits for the owning GID,**
   3. **Permission bits for "world", and**
   4. **The following permission bits: 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:

The process creating the object defines the creator, owner and group based on the effective UID of the current process. Access of a process to an IPC object is allowed, if one of the following rules hold (the order of the rules is applicable on a first-match basis):

- **For signals, the effective UID of the sending process must match the effective UID of the receiving process for allowing the signal transmission; or**
- **The subject's effective UID is equal to the creator UID or owning UID and and the requested type of access is within the permission bits defined for the owning UID; or**
- **The subject's effective GID or one of the subject's supplemental GIDs is equal to the owning GID and the requested type of access is within the permission bits defined for the owning GID; or**

- **The requested type of access is within the permission bits defined for “world”.**

FDP_ACF.1.3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: **all operations except execute operations are allowed for the subject with the effective UID of zero**.

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to named objects based on the following rules: **none.**

6.1.3.4 Security Attribute Based Access Control (at and cron job queues) (FDP_ACF.1(4))

FDP_ACF.1.1 The TSF shall enforce the **Discretionary Access Control Policy** to *at and cron job queue* objects based on the following:

a) **Subject security attributes: effective UID;**

b) **Object security attributes: owning UID;**

c) **Access control security attributes governing access to at and cron job queues: the TOE enforces hard coded permissions not based on user or administrator-modifiable permission attributes.**

d) **Access control security attributes governing access to at and cron job mechanism: /etc/cron.allow, /etc/cron.deny.**

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

a) **Access to cron job queues: A subject has a specific type access to an object if its effective UID equals to the owner of the job queue.**

b) **Access to at job queues: Only the root user has access to the at job queues.**

c) **Access to the cron mechanism: If the file /etc/cron.allow exists, only usernames mentioned in it are allowed to access the cron job queues. If /etc/cron.allow does not exist, /etc/cron.deny is checked which specifies usernames that are denied to access the cron job queues.**

FDP_ACF.1.3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: **all operations except execute operations are allowed for the subject with the effective UID of zero**.

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to named objects based on the following rules: **none.**

6.1.3.5 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 SFP(s), outside of the TOE.

FDP_ETC.1.2 The TSF shall export the *unlabeled* user data without the user data's associated security attributes ~~.~~ *using the following rules:*

- *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.*

- *Only data with the same sensitivity label as the sensitivity label of the device can be exported using the device.*

6.1.3.6 Export of labeled user data (FDP_ETC.2)

FDP_ETC.2.1 The TSF shall enforce the **Mandatory Access Control Policy** when exporting user data, controlled under the SFP(s), outside of the TOE.

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 TOE, 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 TOE:

a) **When the data is exported to a network device, the security attributes shall be exported with the data using the IKEv1 protocol used to establish an SA which is associated with the label applicable with the SA with the remote peer.**

b) **When the data is exported to a file archive, the security attributes shall be exported with the data using the data archiving application storing the labels for each archived file system object as part of the archive.**

6.1.3.7 Mandatory Access Control Policy (FDP_IFC.2)

FDP_IFC.2.1 The TSF shall enforce the **Mandatory Access Control Policy** on **subjects and objects defined with the Security Policy Model** and all operations that cause that information to flow ~~to and from subjects covered by the SFP~~ *among them*.

FDP_IFC.2.2 The TSF shall ensure that all operations that cause any information in the TOE to flow ~~to and from any subject~~ *between subjects and objects* in the TOE are covered by an information flow control SFP.

6.1.3.8 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~~ *object* security attributes:

a) **Subject security attributes:**

   1. **Sensitivity label of the subject consisting of 256 site-definable hierarchical levels and a set of 1024 site-definable non-hierarchical categories;**
   2. **The sensitivity label of the object containing the information.**

b) **Object security attributes:**

   1. **Sensitivity label of the object consisting of at least 256 site-definable hierarchical levels and a set of 1024 site-definable non-hierarchical categories;**

FDP_IFF.2.2 The TSF shall permit an information flow between a controlled subject and controlled ~~information~~ *object* 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 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 information flow is between objects, the sensitivity label of the destination object must be greater than or equal to the sensitivity label of the source object.**

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

FDP_IFF.2.4 The TSF shall explicitly authorise an information flow based on the following rules:

a) **MLS-override attributes assigned to a subject allow that subject to perform the operation the MLS-override attribute applies to irrespectively of the sensitivity labels of the subject or object;**

b) **MLS-override attributes assigned to an object allow every subject to perform the operation the MLS-override attribute applies to with that object irrespectively of the sensitivity labels of the subject or object.**

Application Note:

The following MLS override attributes are defined (note that for most of the below-mentioned attributes the TOE also defines a twin-attribute with the same override-capability which is only applicable when additional restrictions are met - the names of these attributes are identical to their corresponding attribute listed below, extended with the suffix "toclr"):

mlsfdshare

The policy disallows the sharing of file descriptors between levels unless the file descriptor is authorized to be shared among levels.

mlsfduse

The policy disallows the sharing of file descriptors between levels unless the process is authorized to shared it among levels.

mlsfiledowngrade

Make specified domain MLS trusted for lowering the level of files.

mlsfileread

Make specified domain MLS trusted for reading from files at higher levels.

mlsfileupgrade

Make specified domain MLS trusted for raising the level of files.

mlsfilewrite

Make specified domain MLS trusted for writing to files at lower levels.

mlsfilewriteinrange

This attribute has the same meaning as mlsfilewritetoclr.

mlsipcread

Make specified domain MLS trusted for reading from System V IPC objects at any level.

mlsipcwrite

Make specified domain MLS trusted for writing to System V IPC objects at any level.

mlsnetread

Make specified domain MLS trusted for reading from sockets at any level.

mlsnetrecvall

Make specified domain MLS trusted for receiving network data from network interfaces or hosts at any level.

mlsnetwrite

Make specified domain MLS trusted for writing to sockets at any level.

mlsnetwriteranged

Same as mlsnetwritetoclr with even more restrictions on the levels of the process and the target object.

mlsprocread

Make specified domain MLS trusted for reading attributes from processes at higher levels like reading capabilities or scheduling information or performing the ptrace operation.

mlsprocsetsl

Make specified domain MLS trusted for setting the level of processes it executes.

mlsprocwrite

Make specified domain MLS trusted for writing to processes at lower levels like sending signals, setting capabilities, setting the SELinux labels for a process in the proc file.

mlsrangetrans

Make specified domain a target domain for MLS range transitions that change the current level.

mlstrustedobject

Make specified object MLS trusted and exclude it from the MLS checks.

privrangetrans

Allow the specified domain to do a MLS range transition that changes the current level.

Note: The MLS policy specifies additional MLS override attributes. However, those do not cover any objects present in the TOE as they are intended for applications using the SELinux policy in addition to the kernel (such as X11 or databases) - none of these applications are installed in the TOE.

FDP_IFF.2.5 The TSF shall explicitly deny an information flow based on the following rules: **no additional rules**.

FDP_IFF.2.6 The TSF shall enforce the following relationships for any two valid ~~information flow control security attributes~~ *sensitivity labels*:

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

- *Sensitivity labels are equal if the hierarchical levels of both labels are equal and the non-hierarchical category sets are identical;*
- *Sensitivity label A is greater than sensitivity label B if the hierarchical level of A is greater than or equal to the hierarchical level of B, and the non- hierarchical category set of A is identical to or a superset of the non- hierarchical category set of B;*
- *Sensitivity labels are incomparable if they are not equal and neither label is greater than the other as defined in 1 and 2 above;*

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

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

6.1.3.9 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 SFP, from outside of the TOE.

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

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

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.

6.1.3.10 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 SFP, from outside of the TOE.

FDP_ITC.2.2 The TSF shall use the *label-related* 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 the security attributes and the *labeled* user data received.

FDP_ITC.2.4 The TSF shall ensure that interpretation of the *label-related* 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 SFP from outside the TOE:

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.**

6.1.3.11 Full 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 allocation to all objects.

6.1.3.12 Full Residual Information Protection of resources (FDP_RIP.3)

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

Application Note: *The subject is represented by the data structures inside the kernel forming a process: all data structures anchored in the task_struct. The user is represented by its attributes defined by FIA_ATD.1.*

6.1.4 Identification and authentication (FIA)

6.1.4.1 Authentication Failures (FIA_AFL_EXT.1)

FIA_AFL_EXT.1.1 The TSF shall detect when an authorized administrator configurable positive integer of consecutive unsuccessful authentication attempts occur related to any authorized user authentication process.

FIA_AFL_EXT.1.2 When the defined number of consecutive unsuccessful authentication attempts has been met or surpassed, the TSF shall:

a) For all administrator accounts, "disable" the account for an authorized administrator configurable time period such that there can be no more than ten attempts per minute.

b) For all other accounts, disable the user logon account until it is re-enabled by the authorized administrator.

c) For all disabled accounts, any response to an authentication attempt given to the user shall not be based on the result of that authentication attempt.

Application Note: *The configuration of this functional aspect is done by modifying the parameters for the PAM modules in the PAM configuration files stored in /etc/pam.d/.*

6.1.4.2 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) unique identifier,

b) group memberships,

c) authentication data,

d) security-relevant roles (see FMT_SMR.~~2~~ *1*),

e) **No security attribute related to cryptographic functions**, and

f) **Sensitivity label range,**

g) **MLS-override attributes.**

Application Note: *The reference to FMT_SMR.2 has been updated to FMT_SMR.1 as the PP does not specify FMT_SMR.2.*

Application Note: *The maintenance of the security relevant role to a user is done implicitly with the maintenance of the group membershipship for each user. Users which are assigned to the group "wheel" are permitted to use the su application which allows the switch to the root account. Only the root account allows the execution of administrative tasks. See the application note provided for FMT_SMR.1 for more details on the definition of the role of the authorized administrators.*

6.1.4.3 Verification of Secrets (FIA_SOS.1)

FIA_SOS.1.1 The TSF shall provide a mechanism to verify that secrets meet **the following quality metric: the probability that a secret can be obtained by an attacker during the lifetime of the secret is less than 2^-20.**

Application Note: *The TOE password change is implemented using the PAM library. The PAM module pam_passwordqc.so allows the specification of the quality of new passwords. The evaluated configuration requires a configuration of the PAM-based password change mechanism that meets the above mentioned criteria.*

Application Note: *The Evaluated Configuration Guide contains configuration suggestions for the password quality mechanism that covers the above mentioned probability. These configuration suggestions assume the worst-case scenario when attacking these settings.*

6.1.4.4 Timing of authentication (FIA_UAU.1)

FIA_UAU.1.1 The TSF shall allow read access to public objects on behalf of the user to be performed before the user is authenticated.

Application Note: *The following public objects are defined for the TOE:*

- *All network and local protocol objects and information that are necessary to allow users to establish a connection to the identification and authentication mechanism (such as TCP, IP, ARP);*
- *All network and local protocol objects and information that provide debugging or status information relevant to the protocol (such as ICMP);*
- *The information presented with the TOE banner as outlined for FTA_TAB.1.*

FIA_UAU.1.2 The TSF shall require each user to be successfully authenticated (i.e., an exact match between the internal representation of the user’s entered data and the stored TSF authentication data) before allowing any other TSF-mediated actions on behalf of that user.

6.1.4.5 Re-authenticating (FIA_UAU.6)

FIA_UAU.6.1 The TSF shall re-authenticate the user when changing authentication data.

Application Note: *The PAM library uses PAM modules which enforce the password changing mechanism. The PAM module of pam_passwordqc requires users to first provide the current password and then the new password. Only when the supplied current password can be successfully verified, the new password is configured. The root user is exempted from this rule - the root user can set any password for any user without providing the current password of that user.*

6.1.4.6 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.

6.1.4.7 Timing of identification (FIA_UID.1)

FIA_UID.1.1 The TSF shall allow read access to public objects on behalf of the user to be performed before the user is identified.

Application Note: *Please see the application note for FIA_UAU.1 about the specification of public objects.*

FIA_UID.1.2 The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

6.1.4.8 User-subject binding (FIA_USB.1)

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

a) **The security attribute identified in FIA_ATD.1a, b, d, and**

b) **the sensitivity label used to enforce the Mandatory Access Control Policy**

.

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 users:

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

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

c) **Upon successful identification and authentication, the supplemental GIDs shall be those specified via the supplemental group membership assignment for the user entry;**

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

Application Note: *The various subject UIDs are all derived from the same numeric UID per user entry stored in the /etc/passwd file.*

Application Note: *The various subject GIDs except the supplemental GIDs are all derived from the same numeric GID per user entry stored in the /etc/passwd file.*

Application Note: *The subject's supplemental GIDs are derived from the username to group name mappings in the /etc/group file. As the TOE only maintains numeric IDs for subjects, the username and the group names need to be converted before instantiating the subject. The username to UID mapping is provided in /etc/passwd and the group name to GID mapping is provided in /etc/group.*

Application Note: *The initial sensitivity label for each user is maintained in /etc/selinux/mls/seusers. The clearance range for users is specified in the files /etc/selinux/mls/users/*.*

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 users:

a) **The effective and filesystem UID of a subject 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 UID of the owning UID of the file storing the program. These newly set effective and filesystem UIDs are used for the DAC permission validation. The real and login UID remain unchanged.**

b) **The effective and filesystem GID of a subject can be changed by the use of an executable with the SETGID bit set. In this case the program is executed with the effective and filesystem GID of the owning GID of the file storing the program. These newly set effective and filesystem GIDs are used for the DAC permission validation. The real GID remains unchanged.**

c) **The real, effective and filesystem UID of a subject can be changed by the use of the set*uid system call family for the calling application. These system calls are restricted to the root user.**

d) **The real, effective and filesystem GID of a subject can be changed by the use of the set*gid system call family for the calling application. These system calls are restricted to the root user.**

e) **The set of supplemental GIDs of a subject can be changed by the use of the setgroups system call for the calling application. This system call is restricted to the root user.**

f) **The sensitivity label of any subject can be changed to a label within the clearance assigned to the effective UID of that subject. This transition is restricted to subjects possessing the mlsprocwrite or mlsprocwritetoclr MLS override attributes.**

Application Note: *The applications "su" allows the calling user to change the filesystem and effective UID either to root or to other users provided the authentication to "su" was successful. The su application uses the SETUID bit with the owning UID of root as well as the set*uid system calls to change to other UIDs before spawning a new shell. As the su application rests on the above mentioned mechanisms, it is not listed as a separate mechanism to modify the calling user's UIDs.*

Application Note: *The mechanism to change the sensitivity label of subjects is implemented by writing the new label to one of the following files: /proc/<PID>/attr/{current|execve|*create} which allow the specification of the sensitivity label for the running process (current), for the process when the execve system call is triggered (execve) or the sensitivity label that is used when create the next object (*create). The same proc files also exist on a per-thread level.*

Application Note: *The login UID is set by the PAM modules by inserting the intended UID into the /proc/<PID>/loginuid file. This file can be written to only by subjects executing with the effective UID of zero (root) and only for the calling process' own loginuid file. However, there is no application* *except the PAM modules which access that proc file which implies that the login UID remains unchanged after login when operating the TOE. Authorized administrators are not intended to access that proc file.*

6.1.5 Security management (FMT)

6.1.5.1 Management of Functions in TSF (FMT_MOF.1)

FMT_MOF.1.1 The TSF shall restrict the ability to disable and enable the audit functions and to specify which events are to be audited (see FAU_SEL.1.1) to the authorized administrators.

6.1.5.2 Management of Security Attributes (for Discretionary and Mandatory Access Control) (FMT_MSA.1(1))

FMT_MSA.1.1 The TSF shall enforce the Discretionary Access Control policy *and Mandatory Access Control Policy* to restrict the ability to change the value of object security attributes *except for the management of object ownership and the object sensitivity label* to authorized administrators and owners of the object.

Application Note: *Refinement prevents clashes with FMT_MSA.1(2) and adds the aspect of the Mandatory Access Control Policy.*

6.1.5.3 Management of Security Attributes (for Object Ownership) (FMT_MSA.1(2))

FMT_MSA.1.1 The TSF shall enforce the Discretionary Access Control policy *and Mandatory Access Control Policy* to restrict the ability to change object ownership *and the object sensitivity label* to authorized administrators.

6.1.5.4 Secure Security Attributes (FMT_MSA.2)

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

Application Note: *This SFR implies that for all security attributes (subject/user security attributes, named object security attributes) the TOE enforces a valid range of input values. If the caller does not provide the value that falls into the allowed range for that attribute, the TOE rejects the value and therefore the modification attempt.*

6.1.5.5 Static attribute initialisation (FMT_MSA.3(1))

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

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

Application Note: *The default value for permission bits is specified with the umask value which specifies the permission bits for newly created objects. This value has an initial setting of 022 or the value specified in /etc/login.defs. Only administrator can cange that initial value. Users can change their umask value at any time. For ACLs, the default ACL is provided for for the root directory which, in case of absence of a default ACL entry is consistent with the umask.*

Application Note: *at and cron job queues have a default permission setting which cannot be influenced at all. Note that the DAC SFRs for those do not refer to FMT_MSA.3(1).*

6.1.5.6 Static attribute initialisation (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 SFP.

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

Application Note: *The default value for sensitivity labels is derived from the sensitivity label of the calling subject. When an object is created, the object sensitivity label can be altered when providing an alternative sensitivity label to /proc/<PID>/attr/*ceate or /proc/<PID>/task/<TID>/attr/*create where PID and TID specify the process ID / thread ID of the subject that is about to create the object.*

6.1.5.7 Security attribute value inheritance (DAC) (FMT_MSA.4(1))

FMT_MSA.4.1 The TSF shall use the following rules to set the value of security attributes *relevant for the Discretionary Access Control Policy*:

a) **The newly created object's owning UID is set to the effective UID of the calling subject;**

b) **The newly created object's owning GID is set to the effective GID of the calling subject with the following exception for file system objects: if the parent directory holding the newly created file system object is marked with the SETGID permission bit, the owning GID of the newly created file system object is set to the owning GID of the parent directory;**

c) **The newly created object's permission bits are derived from the calling subject's umask value by masking out the umask bits from the permission bit set granting full access;**

d) **The newly created object's ACLs are derived from the default ACL specified for the parent directory the newly created file system object is stored in, if existant. Otherwise, no ACL is set.**

6.1.5.8 Security attribute value inheritance (MAC) (FMT_MSA.4(2))

FMT_MSA.4.1 The TSF shall use the following rules to set the value of security attributes *relevant for the Mandatory Access Control Policy*: **The sensitivity label of the newly created object is set to the sensitivity label of the calling subject. If a label value has been supplied to /proc/<PID>/attr/*ceate or /proc/<PID>/task/<TID>/attr/*create for the calling process or thread, that label value is applied for the creation of the next corresponding object.**

6.1.5.9 Management of TSF Data (Audited Events) (FMT_MTD.1(1))

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

Application Note: *This SFR applies to FAU_SEL.1.*

6.1.5.10 Management of TSF Data (Audit Storage) (FMT_MTD.1(2))

FMT_MTD.1.1 The TSF shall restrict the ability to **clear, configure the storage location, delete, query** the **audit storage** to **authorized administrators.**

Application Note: *This SFR applies to FAU_STG.1.*

6.1.5.11 Management of TSF Data (Audit Threshold) (FMT_MTD.1(3))

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

a) threshold of the audit trail when an action is performed;

b) action when the threshold is reached

to **authorized administrators.**

Application Note: *This SFR applies to FAU_STG.3.*

6.1.5.12 Management of TSF Data (Audit Storage Failure) (FMT_MTD.1(4))

FMT_MTD.1.1 The TSF shall restrict the ability to **modify** the **actions to be taken in case of audit storage failure** to **authorized administrators.**

Application Note: *This SFR applies to FAU_STG.4.*

6.1.5.13 Management of TSF Data (Authentication Failure Threshold) (FMT_MTD.1(5))

FMT_MTD.1.1 The TSF shall restrict the ability to **modify** the **threshold for unsuccessful authentication attempts** to **authorized administrators.**

Application Note: *This SFR applies to FIA_AFL_EXT.1.*

6.1.5.14 Management of TSF Data (Authentication Failure Re-enabling) (FMT_MTD.1(6))

FMT_MTD.1.1 The TSF shall restrict the ability to **re-enable** the **authentication to the account subject to authentication failure** to **authorized administrators.**

Application Note: *This SFR applies to FIA_AFL_EXT.1.*

6.1.5.15 Management of TSF Data (for critical cryptographic security parameters) (FMT_MTD.1(7))

FMT_MTD.1.1 The TSF shall restrict the ability to manage the critical cryptographic security parameters and data related to cryptographic configuration to authorized administrators.

Application Note: *This SFR applies to all cryptography-related SFRs.*

6.1.5.16 Management of TSF Data (User Security Attributes) (FMT_MTD.1(8))

FMT_MTD.1.1 The TSF shall restrict the ability to **initialize, modify, delete, read** the **user security attributes** to

a) **Authorized administrators for all operations;**

b) **Users for modifying their own authentication data;**

c) **Users to read all user security attributes except authentication data.**

Application Note: *This SFR applies to FIA_ATD.1, FIA_UAU.1, FIA_UID.1.*

6.1.5.17 Management of TSF Data (Password quality) (FMT_MTD.1(9))

FMT_MTD.1.1 The TSF shall restrict the ability to **modify** the **password quality configuration** to **authorized administrators.**

Application Note: *This SFR applies to FIA_SOS.1.*

6.1.5.18 Management of TSF Data (Label mapping rules) (FMT_MTD.1(10))

FMT_MTD.1.1 The TSF shall restrict the ability to **create, modify, delete** the **sensitivity label mapping rules for interpreting TSF data received from another trusted IT product** to **authorized administrators.**

Application Note: *This SFR applies to FPT_TDC.1.*

6.1.5.19 Management of TSF Data (File system quotas) (FMT_MTD.1(11))

FMT_MTD.1.1 The TSF shall restrict the ability to **modify** the **file system quota for each user** to **authorized administrators.**

Application Note: *This SFR applies to FRU_RSA.1.*

6.1.5.20 Management of TSF Data (Maximum concurrent sessions) (FMT_MTD.1(12))

FMT_MTD.1.1 The TSF shall restrict the ability to **modify** the **maximum number of concurrent interactive sessions per user** to **authorized administrators.**

Application Note: *This SFR applies to FTA_MCS.1.*

6.1.5.21 Management of TSF Data (Session Locking) (FMT_MTD.1(13))

FMT_MTD.1.1 The TSF shall restrict the ability to **modify** the **time interval of user inactivity before the user's session is locked** to **authorized administrators.**

Application Note: *This SFR applies to FTA_SSL.1.*

6.1.5.22 Management of TSF Data (TOE Banner) (FMT_MTD.1(14))

FMT_MTD.1.1 The TSF shall restrict the ability to **modify** the **advisory note and consent warning message regarding unauthorized use of the TOE** to **authorized administrators.**

Application Note: *This SFR applies to FTA_TAB.1.*

6.1.5.23 Revocation (User security attributes) (FMT_REV.1(1))

FMT_REV.1.1 The TSF shall restrict the ability to revoke security attributes associated with the users under the control of the TSF to authorized administrators.

FMT_REV.1.2 The TSF shall enforce the revocation of security-relevant authorizations at the next logon.

Application Note: *User security attributes are stored in the configuration files mentioned in application notes for other SFRs. Authorized administrators are allowed to change these configuration files using administrative applications provided with the TOE. The changes are enforced for a new session when the user affected by the change initiates that new session.*

6.1.5.24 Revocation (Access permissions) (FMT_REV.1(2))

FMT_REV.1.1 The TSF shall restrict the ability to revoke security attributes of named objects *except the sensitivity labels* to owners of the named object and authorized administrators.

FMT_REV.1.2 The TSF shall enforce the revocation of access rights associated with named objects when an access check is made.

Application Note: *Revocation of security attributes for named objects imply the revocation of access granted to users other than the owner of the object. Note that the DAC ownership management (which can be also considered as a form of access revocation) is specified in FMT_MSA.1(2).*

Application Note: *Sensitivity labels cannot be revoked, they can only be modified as defined by FMT_MSA.1(1). This is consistent with the requirement that all subjects and objects must always bear a label. Therefore, this SFR covers the modification of the sensitivity label which may revoke access for subjects or users to objects.*

6.1.5.25 Time-limited authorization (FMT_SAE.1)

FMT_SAE.1.1 The TSF shall restrict the capability to specify an expiration time for authorized user authentication data to the authorized administrator.

Application Note: *The expiration time and the validity of authentication data is maintained as part of /etc/shadow.*

FMT_SAE.1.2 The TSF shall be able to force the associated authorized user to change their authentication information prior to being able to successfully log on after the expiration time has passed.

Application Note: *The enforcement of the expiration and validity time as well as the enforcement of the change of authentication data is provided with the PAM module of pam_unix.so.*

6.1.5.26 Specification of Management Functions (FMT_SMF.1)

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

a) **Management of auditing;**

b) **Management of cryptographic services;**

c) **Management of the access control policy (DAC);**

d) **Management of the information flow control policy (MAC);**

e) **Management of identification and authentication policy;**

f) **Management of user security attributes;**

Application Note: *The given list is kept generic intentionally. This ST specifies one iteration of FMT_MTD.1 per management function required by an SFR. For each FMT_MTD.1 iteration, a corresponding application note refers to the covered SFR(s).*

6.1.5.27 Security Roles (FMT_SMR.1)

FMT_SMR.1.1 The TSF shall maintain the roles:

a) authorized administrator,

b) **authorized user**.

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

Application Note: *Administrative actions can only be performed when the calling subject possesses the effective UID or file system UID of zero (also called the root user). As the account for the root user is disabled for direct logon, authorized administrators are defined as users who are assigned to the "wheel" group. This group allows the use of the "su" application which is the only way to assume the root user capabilities.*

Application Note: *Subjects with the effective UID or file system UID of zero are still restricted by the MAC policy. To perform administrative actions, the administrative user must possess the following privileges:*

- *Root capability;*
- *Invocation of a subject that possesses one or more MLS override attributes to perform operations which are generally denied by the MAC policy;*
- *Invocation of the "newrole" command to switch the sensitivity label for obtaining write access to system configuration files which are protected by a sensitivity label that is not equal to the sensitivity label of subjects. Please note that an additional access control mechanism is enforced in addition to the MAC policy which is completely disregarded in this ST. This additional access control mechanism (called Type Enforcement that is defined with the "strict" SELinux policy which also includes the MAC policy) adds additional restrictions on top of the MAC policy. In order to perform administrative tasks, the newrole application must also be used to switch the subject type and role covered by the Type Enforcement mechanism.*

6.1.6 Protection of the TSF (FPT)

6.1.6.1 Basic Internal TSF Data Transfer Protection (FPT_ITT.1)

FPT_ITT.1.1 The TSF shall protect TSF data from disclosure when it is transmitted between separate parts of the TOE through the use of the TSF-provided cryptographic services: **specified by FCS_COP.1(1)**.

Application Note: *The TOE executes on one physical system without requiring any functions from remote system. Therefore, this SFR is trivially satisfied by the TOE.*

6.1.6.2 TSF Data Integrity Monitoring (FPT_ITT.3)

FPT_ITT.3.1 The TSF shall be able to detect modification and insertion of TSF data transmitted between separate parts of the TOE through the use of the TSF-provided cryptographic services: **specified FCS_COP.1(3)**.

FPT_ITT.3.2 Upon detection of a data integrity error, the TSF shall take the following actions:

a) audit event, and

b) **no other action**.

Application Note: *The TOE executes on one physical system without requiring any functions from remote system. Therefore, this SFR is trivially satisfied by the TOE.*

6.1.6.3 Manual Recovery (FPT_RCV.1)

FPT_RCV.1.1 After a failure or service discontinuity that may lead to a violation of the TSP, the TSF shall enter a maintenance mode where the ability to return the TOE to a secure state is provided.

6.1.6.4 Reliable Time Stamps (FPT_STM.1)

FPT_STM.1.1 The TSF shall be able to provide reliable time stamps.

6.1.6.5 Internal TSF Data Consistency (FPT_TRC_EXT.1)

FPT_TRC_EXT.1.1 The TSF shall ensure that TSF data is consistent between parts of the TOE by providing a mechanism to bring inconsistent TSF data into a consistent state without undue delay.

Application Note: *The TOE executes on one physical system without requiring any functions from remote system. Therefore, this SFR is trivially satisfied by the TOE.*

6.1.6.6 Inter-TSF basic TSF data consistency (FPT_TDC.1)

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

FPT_TDC.1.2 The TSF shall use **sensitivity label mapping rules defined by an authorized administrator** when interpreting the TSF data from another trusted IT product.

6.1.6.7 TSF Testing (FPT_TST_EXT.1)

FPT_TST_EXT.1.1 The TSF shall run a suite of self tests in accordance with FIPS PUB 140-2 during initial start-up (on power on) to demonstrate the correct operation of the cryptographic modules.

Application Note: *The initial start-up or power-on applies to the startup time of the cryptographic service and not the boot process of the entire system. This approach is consistent with the requirements of FIPS 140-2.*

FPT_TST_EXT.1.2 The TSF shall provide authorized administrators with the capability to verify the integrity of stored TSF executable code through the use of the TSF-provided cryptographic services.

Application Note: *The application rpm is provided with the TOE which links with the NSS library to perform the integrity check of file system objects using SHA-256. In addition, this application also checks file system object meta data, such as permissions, ownership, extended attributes holding ACLs and SELinux labels.*

6.1.7 Resource utilisation (FRU)

6.1.7.1 Maximum Quotas (FRU_RSA.1)

FRU_RSA.1.1 The TSF shall enforce maximum quotas of the following resources: portion of shared persistent storage that individual authorized users can use simultaneously.

6.1.8 TOE access (FTA)

6.1.8.1 Basic limitation on multiple concurrent sessions (FTA_MCS.1)

FTA_MCS.1.1 The TSF shall enforce a maximum number of concurrent interactive sessions per user.

FTA_MCS.1.2 The TSF shall allow an authorized administrator to set the maximum number of concurrent interactive sessions per user.

6.1.8.2 TSF-Initiated Session Locking (FTA_SSL.1)

FTA_SSL.1.1 The TSF shall lock an interactive session after an authorized administrator specified time interval of user inactivity by:

a) clearing or overwriting display devices, making the current contents unreadable.

b) disabling any activity of the user’s data access/display devices other than unlocking the session.

FTA_SSL.1.2 The TSF shall require the user to re-authenticate to unlock the session.

6.1.8.3 User-Initiated Locking (FTA_SSL.2)

FTA_SSL.2.1 The TSF shall allow user-initiated locking of the user’s own interactive session by:

a) clearing or overwriting display devices, making the current contents unreadable.

b) disabling any activity of the user’s data access/display devices other than unlocking the session.

FTA_SSL.2.2 The TSF shall require the user to re-authenticate to unlock the session.

6.1.8.4 Default TOE access banners (FTA_TAB.1)

FTA_TAB.1.1 Before establishing a user session, the TSF shall display an authorized-administrator specified advisory notice and consent warning message regarding unauthorized use of the TOE.

6.1.8.5 TOE Access History (FTA_TAH.1)

FTA_TAH.1.1 Upon successful interactive session establishment, the TSF shall display to the authorized user the date and time of that authorized user’s last successful interactive session establishment.

FTA_TAH.1.2 Upon successful interactive session establishment, the TSF shall display to the authorized user the date and time of the last unsuccessful attempt and the number of unsuccessful attempts at interactive session establishment for that user identifier since the last successful interactive session establishment.

FTA_TAH.1.3 The TSF shall not erase the access history information from the authorized user interface without giving the authorized user the opportunity to review the information.

6.2 Security Functional Requirements Rationale

6.2.1 Security requirements coverage

The following table provides a mapping of SFR to the security objectives, showing that each security functional requirement addresses at least one security objective.

Security Functional Requirements	Objectives
FAU_GEN.1	O.AUDIT_GENERATION
FAU_GEN.2	O.AUDIT_GENERATION
FAU_SAR.1	O.AUDIT_REVIEW
FAU_SAR.2	O.AUDIT_PROTECTION
FAU_SAR.3	O.AUDIT_REVIEW
FAU_SEL.1	O.AUDIT_GENERATION
FAU_STG.1	O.AUDIT_PROTECTION
FAU_STG.3	O.AUDIT_REVIEW
FAU_STG.4	O.AUDIT_REVIEW
FCS_BCM_EXT.1	O.CRYPTOGRAPHIC_SERVICES
FCS_CKM.1(1)	O.CRYPTOGRAPHIC_SERVICES

Security Functional Requirements	Objectives
FCS_CKM.1(2)	O.CRYPTOGRAPHIC_SERVICES
FCS_CKM.4	O.CRYPTOGRAPHIC_SERVICES
FCS_COA_EXT.1	O.CRYPTOGRAPHIC_SERVICES
FCS_COP.1(1)	O.CRYPTOGRAPHIC_SERVICES
FCS_COP.1(2)	O.CRYPTOGRAPHIC_SERVICES
FCS_COP.1(3)	O.CRYPTOGRAPHIC_SERVICES
FCS_RBG_EXT.1	O.CRYPTOGRAPHIC_SERVICES
FDP_ACC.2	O.ACCESS, O.DISCRETIONARY_ACCESS, O.PROTECT
FDP_ACF.1(1)	O.ACCESS, O.DISCRETIONARY_ACCESS, O.PROTECT
FDP_ACF.1(2)	O.ACCESS, O.DISCRETIONARY_ACCESS, O.PROTECT
FDP_ACF.1(4)	O.ACCESS, O.DISCRETIONARY_ACCESS, O.PROTECT
FDP_ETC.1	O.MANDATORY_ACCESS
FDP_ETC.2	O.MANDATORY_ACCESS
FDP_IFC.2	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.PROTECT, O.RESIDUAL_INFORMATION
FDP_RIP.3	O.PROTECT, O.RESIDUAL_INFORMATION
FIA_AFL_EXT.1	O.ACCESS
FIA_ATD.1	O.ACCESS, O.MANDATORY_ACCESS
FIA_SOS.1	O.PROTECT, O.USER_AUTHENTICATION
FIA_UAU.1	O.USER_AUTHENTICATION

Security Functional Requirements	**Objectives**
FIA_UAU.6	O.USER_AUTHENTICATION
FIA_UAU.7	O.PROTECT
FIA_UID.1	O.USER_IDENTIFICATION
FIA_USB.1	O.AUDIT_GENERATION, O.DISCRETIONARY_ACCESS, O.MANDATORY_ACCESS
FMT_MOF.1	O.MANAGE
FMT_MSA.1(1)	O.DISCRETIONARY_USER_CONTROL, O.MANAGE, O.MANDATORY_ACCESS
FMT_MSA.1(2)	O.DISCRETIONARY_USER_CONTROL, O.MANAGE
FMT_MSA.2	O.PROTECT
FMT_MSA.3(1)	O.DISCRETIONARY_ACCESS, O.MANAGE
FMT_MSA.3(2)	O.MANDATORY_ACCESS
FMT_MSA.4(1)	O.ACCESS
FMT_MSA.4(2)	O.MANDATORY_ACCESS
FMT_MTD.1(1)	O.MANAGE
FMT_MTD.1(2)	O.MANAGE
FMT_MTD.1(3)	O.MANAGE
FMT_MTD.1(4)	O.MANAGE
FMT_MTD.1(5)	O.MANAGE
FMT_MTD.1(6)	O.MANAGE
FMT_MTD.1(7)	O.MANAGE
FMT_MTD.1(8)	O.MANAGE
FMT_MTD.1(9)	O.MANAGE
FMT_MTD.1(10)	O.MANAGE
FMT_MTD.1(11)	O.MANAGE
FMT_MTD.1(12)	O.MANAGE
FMT_MTD.1(13)	O.MANAGE
FMT_MTD.1(14)	O.MANAGE

Security Functional Requirements	**Objectives**
FMT_REV.1(1)	O.ACCESS, O.MANAGE
FMT_REV.1(2)	O.ACCESS, O.DISCRETIONARY_USER_CONTROL, O.MANAGE, O.PROTECT
FMT_SAE.1	O.MANAGE
FMT_SMF.1	O.MANAGE
FMT_SMR.1	O.ADMIN_ROLE
FPT_ITT.1	O.REFERENCE_MONITOR
FPT_ITT.3	O.REFERENCE_MONITOR, O.TSF_CRYPTOGRAPHIC_INTEGRITY
FPT_RCV.1	O.RECOVERY, O.REFERENCE_MONITOR
FPT_STM.1	O.AUDIT_GENERATION
FPT_TRC_EXT.1	O.ACCESS, O.RECOVERY, O.REFERENCE_MONITOR
FPT_TDC.1	O.MANDATORY_ACCESS
FPT_TST_EXT.1	O.CORRECT_TSF_OPERATION
FRU_RSA.1	O.RESOURCE_SHARING
FTA_MCS.1	O.ACCESS, O.RESOURCE_SHARING
FTA_SSL.1	O.ACCESS, O.USER_AUTHENTICATION
FTA_SSL.2	O.ACCESS, O.USER_AUTHENTICATION
FTA_TAB.1	O.DISPLAY_BANNER
FTA_TAH.1	O.ACCESS_HISTORY

Table 8: Mapping of security functional requirements to security objectives

6.2.2 Security requirements sufficiency

The following rationale provides justification for each security objective for the TOE, showing that the security functional requirements are suitable to meet and achieve the security objectives:

Security objectives	Rationale
O.ACCESS	The TOE must protect itself and the resources it controls from unauthorized access. FDP_ACC.2 enforces the Discretionary Access Control (DAC) policy on all subjects and all named objects and all operations among them. The DAC policy specifies the access rules between all subjects and all named objects controlled by the TOE. While authorized users are trusted to some extent, this requirement ensures only authorized access is allowed to named objects. All iterations of FDP_ACF.1 specify the DAC policy rules that will be enforced by the TSF and determines if an operation among subjects and named objects is allowed. Furthermore, it specifies the rules to explicitly authorize or deny access to a named object based upon security attributes. The rules for the default security attributes for newly created objects are specified in FMT_MSA.4(1). FIA_AFL_EXT.1 provides a detection mechanism for unsuccessful authentication attempts. The requirement enables an authorized administrator configurable threshold that prevents unauthorized users from gaining access to authorized user's account by guessing authentication data. This mechanism prevents access by either disabling the targeted account. Thus, limiting an unauthorized user's ability to gain unauthorized access to the TOE. FIA_ATD.1 defines the attributes of users, including a userid that is used by the TOE to determine a user's identity and enforce what type of access the user has to the TOE (e.g., the TOE associates a userid with any role(s) they may assume). FMT_REV.1(1) ensures that the authorized administrator has the ability to revoke security attributes to a specific user. This revocation is immediate and helps authorized administrators control the ability of authorized users to log in or perform privileged operations. FMT_REV.1(2) ensures that the authorized administrator and owners of named objects have the ability to revoke security attributes to a specific user. This revocation occurs when an access check is made and helps authorized administrators and owners control the ability of users accessing named objects. FPT_TRC_EXT.1 ensures that the TSF data is consistent between parts of the TOE by providing a mechanism to bring inconsistent TSF data into a consistent state in a timely manner. Such data may become inconsistent if an internal channel between parts of the TOE becomes inoperative or in the case of a distributed TOE, this can occur when parts become disabled, network connections are broken, and so on. The ability to ensure that the TSF data is consistent, between parts of the TOE, affords the TOE the ability to maintain the security policies current throughout all parts of the TOE and limits the opportunity of an outdated security policy to be enforced on parts of the TOE that may be permitting unauthorized access to the TOE and its resources.

Security objectives	Rationale
	FTA_MCS.1 is used to limit the access of users to the TSF. By limiting the number of concurrent interactive sessions, an additional level of controlling access to resources is imposed on users of the TSF. FTA_SSL.1 is used to prevent unauthorized access to the TOE and its resources when an interactive session is left unattended. This requirement ensures that the interactive session will lock by making the visible contents unreadable after a specified time interval of session inactivity. The authorized user needs to re-authenticate to unlock his session. FTA_SSL.2 is used to ensure that unauthorized access to the TOE and its resources when an interactive session is left unattended. It enables the authorized user to lock his interactive session before leaving the session unattended. This eliminates any chance for any user to acquire unauthorized access to an unattended session because there is no time interval of inactivity before the session is locked. The authorized user needs to re-authenticate to unlock his session.
O.ACCESS_HISTORY	FTA_TAH.1 is used to provide information about previous interactive sessions (i.e., date and time). This information is displayed to the authorized user upon each successful interactive session establishment. This requirement gives the authorized users the ability to verify their last successful interactive session and thus, is a means for determining if the previous successful interactive session establishment was authorized or not.
O.ADMIN_ROLE	The TOE must maintain roles to isolate administrative actions. FMT_SMR.1 ensures that a minimum of an administrative role be maintained.
O.AUDIT_GENERATION	FAU_GEN.1 defines the set of events that the TOE must be capable of recording. This requirement ensures that the authorized administrator has the ability to audit any security relevant event that takes place in the TOE. This requirement also defines the information that must be contained in the audit record for each auditable event. There is a minimum of information that must be present in every audit record and this requirement defines that, as well as the additional information that must be recorded for each auditable event. This requirement also places a requirement on the level of detail that is recorded on any additional security functional requirements an ST author adds to this PP. FAU_GEN.2 ensures that the audit records associate a user identity with the auditable event. The association is accomplished using the userid of the authorized user. FAU_SEL.1 allows the authorized administrator to configure which auditable events will be recorded in the audit trail. This provides the administrator with the flexibility in recording only those events that are deemed necessary by site policy, thus reducing the amount of resources consumed by the audit mechanism. FIA_USB.1 plays a role is satisfying this objective by requiring a binding of security attributes associated with users that are authenticated with the subjects that represent them in the TOE. This only applies to

Security objectives	Rationale
	authenticated users, since the identity of unauthenticated users cannot be confirmed. Therefore, the audit trail may not always have the proper identity of the user that causes an audit record to be generated (e.g., an attacker/user providing another user's user identifier). FPT_STM.1 ensures that the time stamps used to create the audit records are reliable. The time and date included in the time stamp is crucial when generating the audit information to ensure accountability.
O.AUDIT_PROTECTION	The audit trail must be protected so that only authorized users and authorized administrators may access it or delete it. FAU_SAR.2 ensures that only authorized users have read access to audit information and FAU_STG.1 ensures that audit information is not modified and protects it from unauthorized deletions.
O.AUDIT_REVIEW	FAU_SAR.1 provides the ability for an authorized administrator to efficiently review audit records. This requirement also mandates the audit information be presented in a manner that is suitable for the administrators to interpret the audit trail. FAU_SAR.3 complements FAU_SAR.1 by providing the administrators the flexibility to specify criteria that can be used to search or sort the audit records residing in the audit trail. FAU_SAR.3 requires the administrators be able to establish the audit review criteria based on a user and identifier, date and time, so that the actions of a user can be readily identified and analyzed. Allowing the administrators to perform searches or sort the audit records based on dates, times, type of events, and success and failure of these events, provides the capability to extract the user activity to what is pertinent at that time in order facilitate the administrator's review. It is important to note that the intent of sorting in this requirement is to allow the administrators the capability to organize or group the records associated with a given criteria. FAU_STG.3 allows the authorized administrator to be alerted of the possible audit data loss if the audit trail exceeds an authorized administrator selectable, pre-defined limit. FAU_STG.4 ensures that all audited events are audited. If the audit trail cannot be written to any more, the system prevents actions that would require the generation of an audit entry.
O.CORRECT_TSF_OPERATION	The test mechanisms defined by FPT_TST_EXT.1 cover: ● The FIPS 140-2 compliant known-answer tests of the cryptographic mechanisms every time when these cryptographic mechanisms are initialized. ● The integrity verification of all TSF binaries and TSF data based on cryptographic mechanisms.
O.CRYPTOGRAPHIC_SERVICES	Baseline cryptographic services are provided in the TOE by FIPS PUB 140-2 compliant modules [FCS_BCM_EXT.1]. Specific functional requirements in the area of cryptographic operations address data encryption and decryption [FCS_COP.1 (1)]; cryptographic signatures [FCS_COP.1 (2)]; cryptographic hashing [FCS_COP.1 (3)]; random number generation [FCS_RBG_EXT.1]; and supporting key management services

Security objectives	Rationale
	[FCS_CKM.1(1), FCS_CKM.1(2), FCS_CKM.4]. These TOE requirements support cryptographic services that can be called upon by the TOE itself, or by TOE authorized users and/or user applications [FCS_COA_EXT.1].
O.DISCRETIONARY_ACCESS	Access to TOE resources is determined by the Discretionary Access Control policy. FDP_ACC.2 ensures that the Discretionary Access Control policy is enforced on all subjects and all named objects and all operations between them. All iterations of FDP_ACF.1 define the Discretionary Access Control rules to determine if any operation between subjects and named objects is allowed. These rules are based on the identity of the users and their group memberships. FIA_USB.1 defines the associations between user security attributes and subjects acting on behalf of that user by which policy decisions are based upon. FMT_MSA.3(1) ensures that the TOE provides protection by default for all named objects at creation time. This may allow authorized users to explicitly specify the desired access controls upon the object at its creation, provided that there is no window of vulnerability through which unauthorized access may be gained to newly-created objects.
O.DISCRETIONARY_USER_CONTROL	To allow authorized users to specify which resources may be accessed, the TOE must provide the ability for object security attributes to be changed and revoked. FMT_MSA.1(1) and FMT_MSA.1(2) restrict the ability to change the value of object security attributes to authorized administrators and owners of objects. FMT_REV.1(2) restricts the ability to revoke security attributes of named objects to authorized administrators and owners of these objects.
O.DISPLAY_BANNER	Before identification and authentication and the establishment of a user session, the TOE allows limited access by any potential users of the system in order to convey warnings and agreements for system use. Through this limited access before establishing a user session, the TSF displays an authorized, administrator-specified advisory notice and consent warning message regarding unauthorized use of the TOE [FTA_TAB.1]. In typical applications a user who continues session establishment procedures (including their successful identification and authentication) after display of the notice and warning banner effectively acknowledges the banner content and consents to the stated conditions. This banner of information can be critical in supporting legal actions related to the use of the TOE.
O.MANAGE	In a variety of ways the TOE supports authorized administrators in the management of security functions, security attributes and data while also restricting unauthorized use. For example, the TOE provides for and restricts the following actions to authorized administrators only (except where specifically noted): • Disable and enable the audit functions, and specify which events are audited [FMT_MOF.1(1)].

Security objectives	Rationale
	• Change the value of object security attributes. (Object owner is also allowed to perform this action.) [FMT_MSA.1(1), FMT_MSA.1(2)]. • Provide restrictive default values for security attributes, and specify alternative initial values to override the default values when an object or information is created. [FMT_MSA.3(1)]. • The management aspects of the individual SFRs are specified with all iterations of FMT_MTD.1. • Revoke security attributes associated with the users within the TSC. [FMT_REV.1 (1)]. • Revoke security attributes of named objects within the TSC. (Object owner is also allowed to perform this action.) [FMT_REV.1 (2)]. • Specify an expiration time for authorized user authentication data. [FMT_SAE.1]. FMT_SMF.1 provides a list of the management functions specified in this PP and is required as a dependency for the management functions.
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.2]. 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(1)]. Protection of named objects must be continuous, starting from object creation [FMT_MSA.3(2)]. The default label for newly created objects is specified with [FMT_MSA.4(2)].
O.PROTECT	O.PROTECT requires mechanisms be provided by the TOE to protect user data and resources. FIA_SOS.1 prescribes the maximum probability for guessing authentication data that must be satisfied. If a user can’t authenticate, he or she will not have the ability to access user data and resources. FIA_UAU.7 ensures that no feedback that affects the ability of users to circumvent the authentication mechanism is presented during the authentication process. The TOE is allowed to provide information that would allow the user to use the authentication mechanism in a correct manner (e.g., press CTRL-ALT-DELTE, slide card quickly, center your finger and press firmly, speak louder and slowly), but not provide information that may allow alteration to their presentation that would thwart the mechanism. FMT_MSA.2 ensures that only valid configuration values are accepted for security attributes, supporting the protection of the TSF by avoiding mis-configuration.

Security objectives	Rationale
	To protect user data and resources, FDP_ACC.2, all iterations of FDP_ACF.1, and FMT_REV.1(2) require a Discretionary Access policy and rules that ensures the correct access to named objects by subjects acting on behalf of users. To ensure that user data is not disclosed before a resource is reused, FDP_RIP.2 and FDP_RIP.3 ensure that the shared memory and operating system controlled files as well as resours are not available to another user thus protecting the user data.
O.RECOVERY	FPT_RCV.1 ensures that the system enters a maintenance mode allowing the system to be returned to a secure state after a failure or service discontinuity. In a secure state, all security policies are enforced. FPT_TRC_EXT.1 provides a mechanism to bring the TOE into a consistent state. TSF data may become inconsistent if an internal channel between parts of the TOE becomes inoperative or in the case of a distributed TOE, this can occur when parts become disabled, network connections are broken, and so on. The ability to ensure that the TSF data is consistent, between parts of the TOE, provides the TOE the ability to maintain the security policies current throughout all parts of the TOE and limits the opportunity of an outdated security policy to be enforced on parts of the TOE that may be permitting unauthorized access to the TOE and its resources. This requirement provides the mechanisms to ensure that upon reconnection, the TSF portions will become in sync over a reasonable time period.
O.RESIDUAL_INFORMATION	FDP_RIP.2 as well as FDP_RIP.3 are used to ensure the contents of resources are not available to subjects or users other than those explicitly granted access to the data.
O.RESOURCE_SHARING	This objective requires mechanisms to prevent authorized users (or software unknowingly acting on their behalf) from exhausting important resources controlled by the TOE in a manner that adversely impacts other users or programs. TOE is required to enforce a limit on the amount of resource a given authorized user may successfully be granted. The resources that are controlled are: CPU time, disk space, system memory, and user accounts. FRU_RSA.1 is intended to enforce the notion that a single authorized user may only be allocated a "preset maximum" amount of resource. The requirement only covers persistent storage to offer confidence that entities executing on the TOE are not “starved for persistent storage” and will be allowed to initiate and complete execution. FTA_MCS.1 identifies user accounts as a system resource that could be exhausted (through multiple concurrent "logons" of a single individual). The requirement mandates that the administrator be able to limit the number of concurrent logon sessions by a single user. This ensures that a single individual could not mount a denial-of-service attack using multiple sessions as launching points. Resources (e.g., memory contained on the network card) that are not covered by the above are subject to denial of service attacks. Denial-of-service attacks of these resources should be addressed via other mechanisms such as redundant hardware.

Security objectives	Rationale
O.REFERENCE_MONITOR	This objective requires the protection of the TSF (and its data) from external interference, tampering or inappropriate disclosure by mandating that the TSF create and maintain a domain for its execution. Domain is defined as the logical area that the TSF provides for itself in which to operate. Common mechanisms include hardware execution domains (e.g., processor execution rings as well as other isolation mechanisms that protect TSF data when it is in transit to other TSF components.) The requirements that implement this objective fall into two categories. The first category mandates mechanisms to implement a secure domain for execution. The second category mandates that if the TSF (for some reason) moves into an unknown or unconnected state, that it has a way to recover to a known or connected state. This ensures that the TSF can continue to protect itself even after unexpected interruptions. Requirements included in the first category are FPT_ITT.1 and FPT_ITT.3 (in addition several assurance requirements). The FPT_ITT requirements protect TSF data in transmission between remote portions of the TSF and also require that mechanisms be in place to protect against man-in-the-middle replay attacks that could attempt to interfere with the TSF policy being enforced. Requirements included in the second category are FPT_RCV.1 and FPT_TRP_EXT.1. FPT_RCV.1 is used to ensure that the TSF offers a mechanism to recover from a failed state by mandating that the TSF provide maintenance mode from which to re-initiate (or establish) a known (secure) state. This ensures that once the TSF has established a domain for its own execution it can always return to that state with confidence that this domain continues to be present. FPT_TRP_EXT.1 is used to address distributed TSFs and the fact that portions of these TSF may become disconnected over time. A disconnected portion of the TSF does not always suggest an insecure state or discontinuity of service (referenced in FPT_RCV.1). Instead, this requirement addresses the situation when a portion of a distributed TSF is disconnected from the rest of the TSF (with both pieces continuing service). Specifically, it requires that there be mechanisms provided by the TSF to ensure that upon reconnection, the TSF portions will become in sync over a reasonable time period.
O.TSF_CRYPTOGRAPHIC_INTEGRITY	This objective requires the TOE to provide cryptography that must be used to protect TSF data as it is transmitted between parts of a physically distributed TOE. FPT_ITT.3 requires that the TSF shall be able to use encryption to detect modification, insertion and replay of TSF data transmitted between separate parts of the TOE.
O.USER_AUTHENTICATION	FIA_UAU.1 plays a role in satisfying this objective by ensuring that every user is authenticated before the TOE performs any TSF-mediated actions on behalf of that user. FIA_UAU.6 ensures that the authorized user changing his authentication data re-authenticates before he or she is allowed to proceed.

Security objectives	Rationale
	To verify the claimed identity of an authorized user, FIA_SOS.1 prescribes the metrics that must be satisfied. It provides the mechanism that will verify the secret for user authentication. The PP authors intentionally did not dictate that a password mechanism be required and allowed for other types of authentication mechanisms (e.g. a PIN, Token). In any case, FIA_SOS.1 requires that the authentication mechanism provide the ability for authorized users to have a “secret” up to 16 characters in length, consisting of any combination of upper and lower case letters, numbers, and punctuation. FTA_SSL.1 and FTA_SSL.2 ensure that the authorized user authenticates him or herself before accessing a locked interactive session. This eliminates any chance for any user to acquire unauthorized access to an unattended session. Active interactive sessions may be locked by a user or after a specified time interval of user inactivity configured by an authorized administrator.
O.USER_IDENTIFICATION	FIA_UID.1 plays a role in satisfying this objective by ensuring that every user is identified before the TOE performs any TSF-mediated actions on behalf of that user. It also allows for the specification of a list of public objects that users are allowed read access before the user is identified.

Table 9: Security objectives for the TOE rationale

6.2.3 Security requirements dependency analysis

The following table demonstrates the dependencies of SFRs modeled in CC Part 2 and how the SFRs for the TOE resolve those dependencies:

Security Functional Requirement	Dependencies	Resolution
FAU_GEN.1	FPT_STM.1	FPT_STM.1
FAU_GEN.2	FAU_GEN.1	FAU_GEN.1
	FIA_UID.1	FIA_UID.1
FAU_SAR.1	FAU_GEN.1	FAU_GEN.1
FAU_SAR.2	FAU_SAR.1	FAU_SAR.1
FAU_SAR.3	FAU_SAR.1	FAU_SAR.1
FAU_SEL.1	FAU_GEN.1	FAU_GEN.1
	FMT_MTD.1	The dependency on FMT_MTD.1 is resolved by FMT_MOF.1(1) specifying the management aspect applicable to this SFR.
FAU_STG.1	FAU_GEN.1	FAU_GEN.1

Security Functional Requirement	Dependencies	Resolution
FAU_STG.3	FAU_STG.1	FAU_STG.1
FAU_STG.4	FAU_STG.1	FAU_STG.1
FCS_BCM_EXT.1	No dependencies.	
FCS_CKM.1(1)	[FCS_CKM.2 or FCS_COP.1]	FCS_CKM.1(1)
	FCS_CKM.4	FCS_CKM.4
		FCS_RBG_EXT.1
FCS_CKM.1(2)	[FCS_CKM.2 or FCS_COP.1]	FCS_CKM.1(2)
	FCS_CKM.4	FCS_CKM.4
		FCS_RBG_EXT.1

FCS_CKM.4	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_CKM.1(1)
		FCS_CKM.1(2)

FCS_COA_EXT.1	FCS_BCM_EXT.1	FCS_BCM_EXT.1
FCS_COP.1(1)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_CKM.1(1)
	FCS_CKM.4	FCS_CKM.4
FCS_COP.1(2)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_CKM.1(2)
	FCS_CKM.4	FCS_CKM.4
FCS_COP.1(3)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	The hash mechanisms listed in this SFR do not require any cryptographic keys. This implies that also no cryptographic keys need to be generated and/or destroyed for these hash mechanisms, rendering the dependencies to FCS_CKM.1 as not applicable.
	FCS_CKM.4	The hash mechanisms listed in this SFR do not require any cryptographic keys. This implies that also no cryptographic keys need to be generated and/or destroyed for these hash mechanisms, rendering the dependencies to FCS_CKM.4 as not applicable.
FCS_RBG_EXT.1	FCS_BCM_EXT.1	FCS_BCM_EXT.1

FDP_ACC.2	FDP_ACF.1	FDP_ACF.1(1)
		FDP_ACF.1(2)
		FDP_ACF.1(4)

Security Functional Requirement	Dependencies	Resolution
FDP_ACF.1(1)	FDP_ACC.1	FDP_ACC.2
	FMT_MSA.3	FMT_MSA.3(1)
FDP_ACF.1(2)	FDP_ACC.1	FDP_ACC.2
	FMT_MSA.3	FMT_MSA.3(1)
FDP_ACF.1(4)	FDP_ACC.1	FDP_ACC.2
	FMT_MSA.3	Due to the fact that the permission settings for at and cron job queues are hard-coded without the possibility to alter them for new or existing job queues, the management aspect is not applicable.
FDP_ETC.1	[FDP_ACC.1 or FDP_IFC.1]	FDP_IFC.2
FDP_ETC.2	[FDP_ACC.1 or FDP_IFC.1]	FDP_IFC.2
FDP_IFC.2	FDP_IFF.1	FDP_IFF.2
FDP_IFF.2	FDP_IFC.1	FDP_IFC.2
	FMT_MSA.3	FMT_MSA.3(2)
FDP_ITC.1	[FDP_ACC.1 or FDP_IFC.1]	FDP_IFC.2
	FMT_MSA.3	FMT_MSA.3(2)
FDP_ITC.2	[FDP_ACC.1 or FDP_IFC.1]	FDP_IFC.2
	[FTP_ITC.1 or FTP_TRP.1]	The assumption A.CONNECT requires a protected network ensuring that the the communication channels between the TOE and the remote peer is trusted excluding the requirement for FTP_ITC.1.
	FPT_TDC.1	FPT_TDC.1
FDP_RIP.2	No dependencies.	
FDP_RIP.3	No dependencies.	
FIA_AFL_EXT.1	FIA_UAU.1	FIA_UAU.1
FIA_ATD.1	No dependencies.	
FIA_SOS.1	No dependencies.	
FIA_UAU.1	FIA_UID.1	FIA_UID.1
FIA_UAU.6	No dependencies.	

Security Functional Requirement	Dependencies	Resolution
FIA_UAU.7	FIA_UAU.1	FIA_UAU.1
FIA_UID.1	No dependencies.	
FIA_USB.1	FIA_ATD.1	FIA_ATD.1
FMT_MOF.1	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MSA.1(1)	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.2FDP_IFC.2
	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MSA.1(2)	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.2
	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MSA.2	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.2

	FMT_MSA.1	FMT_MSA.1(1)
		FMT_MSA.1(2)

	FMT_SMR.1	FMT_SMR.1

FMT_MSA.3(1)	FMT_MSA.1	FMT_MSA.1(1)
		FMT_MSA.1(2)

	FMT_SMR.1	FMT_SMR.1
FMT_MSA.3(2)	FMT_MSA.1	FMT_MSA.1(1)
	FMT_SMR.1	FMT_SMR.1
FMT_MSA.4(1)	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.2
FMT_MSA.4(2)	[FDP_ACC.1 or FDP_IFC.1]	FDP_IFC.2
FMT_MTD.1(1)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(2)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(3)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1

Security Functional Requirement	Dependencies	Resolution
FMT_MTD.1(4)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(5)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(6)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(7)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(8)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(9)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(10)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(11)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(12)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(13)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(14)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_REV.1(1)	FMT_SMR.1	FMT_SMR.1
FMT_REV.1(2)	FMT_SMR.1	FMT_SMR.1
FMT_SAE.1	FMT_SMR.1	FMT_SMR.1
	FPT_STM.1	FPT_STM.1
FMT_SMF.1	No dependencies.	
FMT_SMR.1	FIA_UID.1	FIA_UID.1

Security Functional Requirement	Dependencies	Resolution
FPT_ITT.1	No dependencies.	FCS_COP.1(1)
FPT_ITT.3	FPT_ITT.1	FPT_ITT.1
		FCS_COP.1(3)
FPT_RCV.1	AGD_OPE.1	AGD_OPE.1
FPT_STM.1	No dependencies.	
FPT_TRC_EXT.1	No dependencies.	
FPT_TDC.1	No dependencies.	

FPT_TST_EXT.1	FCS_COP.1	FCS_COP.1(1)
		FCS_COP.1(2)
		FCS_COP.1(3)

	FCS_RBG_EXT.1	FCS_RBG_EXT.1
FRU_RSA.1	No dependencies.	
FTA_MCS.1	FIA_UID.1	FIA_UID.1
FTA_SSL.1	FIA_UAU.1	FIA_UAU.1
FTA_SSL.2	FIA_UAU.1	FIA_UAU.1
FTA_TAB.1	No dependencies.	
FTA_TAH.1	No dependencies.	

Table 10: TOE SFR dependency analysis

6.3 Security Assurance Requirements

The security assurance requirements for the TOE are the Evaluation Assurance Level 4 components, augmented by ALC_FLR.3, as specified in [CC] part 3. No operations are applied to the assurance components.

6.4 Security Assurance Requirements Rationale

The evaluation assurance level commensurate with the threat environment that is experienced by typical consumers of the TOE. In addition, the evaluation assurance level augmented with ALC_FLR.3 commensurate with the augmented flaw remediation capabilities offered by the developer beyond those required by the evaluation assurance level.

7 TOE Summary Specification

7.1 TOE Security Functionality

The following section explains how the security functions are implemented. The different TOE security functions cover the various SFR classes.

The primary security features of the TOE are:

- Identification and Authentication
- Audit
- Discretionary Access Control
- Mandatory Access Control
- Cryptographic services
- Security Management
- TSF Protection

7.1.1 Identification and Authentication

User identification and authentication in the TOE includes all forms of interactive login (e.g. using the SSH protocol or log in at the local console) as well as identity changes through the su command. These all rely on explicit authentication information provided interactively by a user.

7.1.1.1 Common Identification and Authentication mechanisms

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 TOE provides PAM modules that implement all the security functionality to:

- Provides login control and establishing all UIDs, GIDs and login ID for a subject
- Ensure the quality of passwords
- Enforce limits for accounts (such as the number of maximum concurrent sessions allowed for a user)
- Enforce the change of passwords after a configured time
- Restriction of the use of the root account to certain terminals
- Restriction of the use of the su command
- Set up of the sensitivity label and file system name space

The login processing sets the real, file system effective and login UID as well as the real, effective, file system GID and the set of supplemental GIDs of the subject that is created. It is of course up to the client application usually provided by a remote system to protect the user’s entry of a password correctly (e. g. provide only obscured feedback).

During login processing, the user is shown a banner. After successful authentication, the login time is recorded.

After a successful identification and authentication, the TOE initiates a session for the user and spawns the initial login shell as the first process the user can interact with. The TOE provides a mechanism to lock a session either automatically after a configurable period of inactivity for that session or upon the user's request.

This security function covers the SFRs of FMT_SMR.1, FAU_UAU.1, FAU_UAU.6, FIA_SOS.1, FIA_UID.1, FAU_GEN.1, FIA_AFL_EXT.1, FIA_USB.1, FIA_UAU.7, FTA_MCS.1.

7.1.1.2 User Identity Changing

Users can change their identity (i.e., switch to another identity) using the su command. When switching identities, the real, file system and effective user ID and real, file system and effective group ID are changed to the one of the user specified in the command (after successful authentication as this user).

The primary use of the su command within the TOE 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. In addition the use of the su command to switch to root has been restricted to users belonging to a special group. Users that don't have access to a terminal where root login is allowed and are not member of that special group will not be able to switch their real, file system 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 file system ID are changed to that of the owner of the file containing the program while the real user ID remains that of the caller. The login ID is neither changed by the su command nor by executing a program that has the setuid or setgid bit set as it is used for auditing purposes.

Note: The login ID is not retained for the following special case:

1. User A logs into the system.
2. User A uses su to change to user B.
3. User B now edits the cron job queue to add new jobs. This operation is appropriately audited with the proper login ID.
4. Now when the new jobs are executed as user B, the system does not provide the audit information that the jobs are created by user A.

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

This security function covers the SFR of FIA_USB.1.

7.1.1.3 Authentication Data Management

Each TOE instance 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 TOE instances. As a result the same user may have different user names, 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 TOE instance within the network maintains its own administrative database by making all administrative changes on the local TOE instance. System administration has to ensure that all machines within the network are configured in accordance with the requirements defined in this Security Target.

The file /etc/passwd contains for each user the user's name, the id of the user, an indicator whether the password of the user is valid, the principal group id of the user and other (not security relevant) information. The file /etc/shadow contains for each user a hash of the user's password, the userid, the time the password was last changed, the expiration time as well as the validity period of the password and some other information that are not subject to the security functions as defined in this Security Target. Users are allowed to change their passwords by using the passwd command. This application is able to read and modify the contents of /etc/shadow for the user’s password entry, which would ordinarily be inaccessible to a non-privileged user process (this implies that the TSF does not rely on the strength of the hashing algorithm to protect the passwords). 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.

The time of the last successful logins is recorded in the /var/log/lastlog file.

The TOE displays informative banners before or while users are logging in. The banners can be specified with the files /etc/issue for log ins via the physical console or /etc/issue.net for remote log ins, such as via SSH. When logging into through the physical console, the banner is displayed above the username and password prompt. For logging in via SSH, the banner is displayed to the remote peer before the SSH-session handshake takes place. The remote SSH client will display the banner to the user. When using the provided OpenSSH client, the banner is displayed when the user instructs the OpenSSH client to log into the remote system.

This security function covers the SFRs of FTA_TAH.1, FMT_SMR.1, FTA_TAB.1, FAU_UAU.1, FAU_UAU.6, FIA_SOS.1, FIA_UID.1, FAU_GEN.1, FIA_AFL_EXT.1, FIA_ATD.1, FIA_USB.1, FIA_UAU.7, FTA_MCS.1.

7.1.1.4 User session handling

Sessions can be locked by users voluntarily via the screen application. In addition, the vlock application is started to protect the user's session after a configurable duration of inactivity on that session. To ensure that the session is always locked even when applications take full control of the session, a helper-daemon may be used that controls each session and terminates offending applications.

This security function covers the SFRs of FTA_SSL.1, FTA_SSL.2.

7.1.2 Audit

The Lightweight Audit Framework (LAF) is designed to be an audit system for Linux compliant with the requirements from Common Criteria. LAF is able to intercept all system calls as well as retrieving audit log entries from privileged user space applications. 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.

7.1.2.1 Audit functionality

The kernel interface which provides the means to configure the audit properties is usable only by root users. Only processes possessing the root authority or kernel functions can submit audit records to the kernel which in turn forwards the audit records to the audit daemon. The audit daemon 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, is halted or the audit daemon executes an administrator-specified notification action depending on the configuration of the audit daemon. 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.

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.

The system administrator can define the events to be audited from the overall events that the Lightweight Audit Framework using simple filter expressions. 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 or alternatively a set of user IDs that are not audited.

The system administrator can select files to be audited by adding them to a watch list that is loaded into the kernel.

7.1.2.2 Audit trail

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)
- Sensitivity label of the subject that caused the event

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 time stamp and audit ID to permit easy correlation.

The audit trail is stored in ASCII text. 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
- ausearch - allows selective extraction of records from the audit trail using defined selection criteria
- sort - 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.

The audit trail is stored in files which are accessible by root only.

This security function covers the SFRs of FAU_GEN.1, FAU_GEN.2, FAU_SEL.1, FPT_STM.1. FAU_STG.1, FAU_SAR.1, FAU_SAR.3, FAU_STG.3, FAU_STG.4.

7.1.3 Discretionary Access Control

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

A subject with a file system user ID of 0 is exempt from all restrictions of the discretionary access control and can perform any action desired. For the execution of a file by root, the permission bit vector of that file must contain at least one execute bit.

DAC provides the mechanism that allows users to specify and control access to objects that they own. DAC attributes are assigned to objects at creation time and remain in effect until the object is destroyed or the object attributes are changed. DAC attributes exist for, and are particular to, each type of named object known to the TOE. DAC is implemented with permission bits and, when specified, ACLs.

The outlined DAC mechanism applies only to named objects which can be used to store or transmit user data. Other named objects are also covered by the DAC mechanism but may be supplemented by further restrictions. These additional restrictions are out of scope for this evaluation. Examples of objects which are accessible to users by cannot be used to store or transmit user data are: virtual file systems externalizing kernel data structures (such as most of procfs, sysfs, binfmt_misc) and process signals.

During creation of objects, the TSF ensures that all residual contents is removed from that object before making it accessible to the subject requesting the creation.

This security function covers FDP_RIP.2 FMT_REV.1(1), FMT_REV.1(2), FMT_MSA.1(1), FMT_MSA.1(2), FMT_MSA.2, FMT_MSA.3(1).

7.1.3.1 Permission bits

The TOE supports standard UNIX permission bits to provide one form of DAC for file system objects in all supported file systems. 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. Also, write access to file system objects marked as immutable is always rejected. The SAVETXT attribute is used for world-writeable temp directories preventing the removal of files by users other than the owner.

Each process has an inheritable “umask” attribute which is used to determine the default access permissions for new objects. It is a bit mask 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. The umask is defined by the administrator in the /etc/login.defs file or 022 by default if not specified.

7.1.3.2 Access Control Lists (ACLs)

The TOE provides support for POSIX type ACLs to define a fine grained access control on a user basis. ACLs are supported for all file system objects stored with the following file systems:

- ext3
- tmpfs

An ACL entry contains the following information:

- A tag type that specifies the type of the ACL entry
- A qualifier that specifies an instance of an ACL entry type
- A permission set that specifies the discretionary access rights for processes identified by the tag type and qualifier

An ACL contains exactly one entry of three different tag types (called the "required ACL entries" forming the "minimum ACL"). The standard UNIX file permission bits as described in the previous section are represented by the entries in the minimum ACL.

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. 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.

7.1.3.3 File system objects

Access to file system objects is generally governed by permission bits. For the ext3 file system, ACLs are supported.

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.

This security function covers FDP_ACC.2, FDP_ACF.1 (1).

7.1.3.4 IPC objects

The TOE implements the following standard types of IPC mechanisms:

- SYSV Shared Memory
- SYSV and POSIX Message Queues
- SYSV Semaphores

Access to the above mentioned IPC mechanisms are governed by UNIX permission bits.

As the IPC objects of UNIX domain socket special files and Named Pipes are represented as file system objects, the access control mechanism covering file system objects are applicable to these IPC mechanisms too.

The TOE maintains IPC object types where each process has its own namespace for that object type: sockets - including network sockets. Access to the socket is only possible by the process whose socket namespace contains the socket reference. Setting of permissions for such objects can be handled using file descriptor passing.

This security function covers FDP_ACC.2, FDP_ACF.1(2).

7.1.3.5 at and cron jobs queues

cron jobs can only be accessed (read/added/modified/deleted) by the owning user. The TOE maintains cron job queues (i.e. the crontab files) for each user. at job queues are accessible to the root user only. Note that each cron job queue is defined with one crontab file.

The root user can always access every cron job queue.

Access to the cron mechanisms can be limited using the /etc/cron.allow and the /etc/cron.deny files. If the allow file exists, only the users specified in these allow file are allowed to access his at or cron job queue, respectively. In case the allow files do not exist, the deny files are analyzed. Only users specified in the deny files are denied access to his at or cron job queue, respectively.

The at or cron jobs are started with the UIDs/GIDs of the creator of the job.

This security function covers FDP_ACC.2, FDP_ACF.1 (4).

7.1.4 Mandatory Access Control

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 non-hierarchical set of categories.

The SELinux security module attaches a “sensitivity label” as part of the security context to the objects defined in Security Policy specification .

Processes are subjects with associated security contexts. When sending signals using the kill system call, the target process behaves like an object.

In addition a process as a subject also has a security context attached. 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 a specific MLS override attribute. Access control is performed based on the sensitivity labels of the process and the object the process interacts with.

When access attempts by a subject onto an object covered by the Discretionary Access Control are performed, the Mandatory Access Control policy is only enforced after the Discretionary Access Control policy allowed the access attempt. In case the Discretionary Access Control policy denies the access attempt, the denial decision is immediately returned to the calling subject.

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 obtain a security context (including a sensitivity label) when they are created in accordance with the rules of the mandatory access control policy.

To support world-writeable directories or home directories for users which can access the system with different labels, the concept of polyinstantiated directories is implemented by the TOE. Polyinstantiation of directories implies that a user's process can only see the file system objects with the same label that his process is assigned with. Considering the purpose of polyinstantiated directories which tries to separate the file system objects of different labels it is clear that polyinstantiation is not relevant for DAC.

This security function covers all SFRs of FDP_IFC.2, FDP_IFF.1, FIA_ATD.1, FIA_USB.1, FMT_MSA.1(1), FMT_MSA.3(2), FMT_MSA.4(2).

7.1.4.1 at and cron jobs queues

The TOE maintains at and cron job queues for each sensitivity label per user and applies the mandatory access control rules when accessing these queues.

Processes spawned by at or cron are assigned the sensitivity label of the creator of the job.

The at program is not a setuid program; therefore, it cannot be executed by regular users. It would be in violation of the evaluated configuration to change the at program to a setuid program.

7.1.4.2 Export/Import of labeled and unlabeled data

The system supports import and export of unlabeled data. When using single level devices, changes in device level must be performed manually by the administrator and are auditable.

An data archiving tool permits import and export of labeled filesystem data when used by administrators by creating archives that preserve label information.

The TOE IPsec implementation allows assigning labels to network objects and enforcing the mandatory access control policy based on those labels.

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.

This security function covers all SFRs of FDP_ETC.1, FDP_ETC.2, FDP_ITC.1, FDP_ITC.2, FPT_TDC.1.

7.1.5 Cryptographic services

The TOE provides the NSS library which is covered by a FIPS 140-2 certificate - other cryptographic services implement the cryptographic mechanisms as asserted by the vendor. The NSS library is used in a FIPS 140-2 compliant mode by the following services:

- Wrapper application - providing users with access to the general-purpose cryptographic services
- TSF integrity check - using the cryptographic services of the NSS library to implement the TSF integrity verification mechanism

7.1.5.1 NSS wrapper application

In the evaluated configuration, any user requesting general services from the NSS library shall only use the provided wrapper application to interact with the cryptographic mechanisms of the NSS library. User applications must not link with the NSS library directly as the proper operation of these services for the caller cannot be guaranteed.

The NSS wrapper allows provides the following services to any caller:

- Generation of symmetric and ECDSA keys
- Destruction of symmetric and ECDSA keys
- Encryption and decryption using the AES cipher
- Signature generation using the ECDSA mechanism
- Message digest generation using the SHA-2 family

The NSS library uses a deterministic random bit generator seeded by /dev/urandom. This file provides access to the kernel-maintained non-blocking entropy pool which is filled based on first and second derivation of the time deltas between the occurrence of selected hardware interrupts. The kernel uses carefully selected hardware interrupt sources to prevent attackers from predicting the entropy. In case the entropy pool runs low on entropy, the kernel applies a deterministic random number generation mechanism utilizing the SHA-1 algorithm until sufficient entropy can be obtained from the interrupt sources.

This security function covers all SFRs of FCS_BCM_EXT.1, FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3), FCS_RBG_EXT.1, FCS_CKM.1(1), FCS_CKM.1(2), FCS_CKM.4, FCS_COA_EXT.1.

7.1.5.2 TSF integrity check

The TOE implements an integrity verification tool which maintains a database with SHA-256 hashes of all TSF binary files as well as the meta data of these files (such as permission bits, owner ship information, modification time, file system object name).

The integrity verification mechanism scans the TSF binary files and other configured files and matches each file with the attributes stored in the database. If an attribute does not match, it generates a warning.

Besides the verification of the integrity of the TSF binary files, the integrity verification mechanism can also be used to update the integrity check database in case file system objects under control of that mechanism are intentionally changed. The database is accessible by root only and can therefore only be updated by an authorized administrator.

To calculate the hash values of files, the integrity verification mechanism uses the NSS library services.

This security function covers all SFRs of FPT_TST_EXT.1.

7.1.6 Security Management

The security management facilities provided by the TOE are usable by authorized users and/or authorized administrators to modify the configuration of TSF. The configuration of TSF are hosted in the following locations:

- Configuration files (or TSF databases)
- Data structures maintained by the kernel and within the kernel memory

The TOE provides applications to authorized users as well as authorized administrators to perform various administrative tasks. These applications are documented as part of the administrator and user guidance. These applications are either used to modify configuration files or to access parameters controlled and enforced by the kernel via kernel-provided interfaces to user space.

Configuration options are stored in different configuration files. These files are protected using the DAC mechanisms against unauthorized access (note that although these files are also covered with MAC protection, that protection is considered to be irrelevant for ensuring information flow control as only the administrator is able to access them to add unspecified information). 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. These configuration files are accessed using applications which are able to interpret the contents of these configuration files. Each TOE instance 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.

To access data structures maintained by the kernel, applications use the kernel-provided interfaces, such as system calls, virtual file systems, netlink sockets, and device files. These kernel interfaces are restricted to authorized administrators or authorized users, if applicable, by either using DAC (for virtual file system objects) or special kernel-internal verification checks for each interface.

The TOE provides security management applications for all security-relevant settings listed throughout this ST, i.e. all FMT_MSA.1 and FMT_MTD.1 iterations.

This security function covers all SFRs mapped to FMT_SMR.1, FMT_MOF.1(1), FMT_MSA.1(1), FMT_MSA.1(2), FMT_MSA.3(1), all iterations of FMT_MTD.1, FMT_REV.1(1), FMT_REV.1(2), FMT_SAE.1, FIA_SOS.1, FIA_UAU.7, FMT_MSA.2, FDP_ACC.2, FDP_ACF.1(1), FDP_RIP.2, FDP_RIP.3.

7.1.7 TSF Protection

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.

Non-kernel TSF software and data are protected by DAC and process isolation mechanisms. In the evaluated configuration, DAC permission settings 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 TSF including the hardware and firmware components are required to be physically protected from unauthorized access. The kernel mediates all access to the hardware mechanisms, 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 is adequately protected using proper DAC permission settings.

7.1.7.1 TSF Invocation Guarantee

All system protected resources are managed by the TSF. Because all TSF data and the associated TSF data structures are protected, these resources can be directly manipulated only by the TSF using defined TSF interfaces. This satisfies the condition that the TSF must be "always invoked" to manipulate protected resources.

Resources managed by the kernel software can only be manipulated while running in kernel mode.

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

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.

7.1.7.2 Kernel

The TOE 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 and 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).

The kernel is shared by all processes, and manages system wide shared resources. It presents the primary programming interface for the TOE 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.

The TOE 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. 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 invoke a user space application which looks for the requested module and loads it using system calls.

7.1.7.3 Trusted Processes

Trusted processes in the TOE 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).

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. 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 the TOE system strictly controls those programs and prohibits that those programs are modified or that programs from untrusted sources are executed with root privileges.

Trusted processes are part of the TSF.

7.1.7.4 Secure failure state

The system provides a single user maintenance mode. The system can be configured to automatically enter single user mode when the self test utility detects a security failure. The self test is performed during boot time.

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.

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

7.1.7.5 Resource limits

The TOE controls the usage of resources by the subjects. Resource limits can be configured by authorized administrators and are enforced by the TSF.

The following resource limits are provided:

- Session limit - The number of concurrent sessions of one user can be limited to an administrator configurable number. The TOE enforces this limit using a PAM module.
- File system quota - The amount of file system storage space usable by one user identified with his user ID can be limited to an administrator configurable limit. That limit is enforced by the kernel.

This security function covers the SFRs of FPT_RCV.1, FRU_RSA.1, FTA_MCS.1.

As the TOE is only executed on one hardware system and does not rely on other systems for enforcing the security functionality, the TOE is considered to be not a distributed system. In that effect, FPT_TRC_EXT.1, FPT_ITT.1, FPT_ITT.3 are not applicable and therefore trivially met by the system architecture.

8 Abbreviations, Terminology and References

8.1 Abbreviations

ACL

Access Control List

API

Application Programming Interface

HTTP

Hypertext Transfer Protocol

SFR

Security Functional Requirement

SSL

Secure Sockets Layer

ST

Security Target

TCP/IP

Transmission Control Protocol / Internet Protocol

TLS

Transport Layer Security

TOE

Target of Evaluation

TSF

TOE Security Functionality

VM

Virtual Machine

VPN

Virtual Private Network

8.2 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.

Authentication Data

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

Authorized Administrator

This term refers to a user in one of the defined administrative roles of a Linux system. The TOE associates the user with the UID of zero and named "root" with administrative authorities. Effectively, the UID zero is assigned with all Linux capabilities known to the Linux kernel. Every user who is allowed to log on as that root user or to switch their UID to the root user is considered an authorized administrator. In addition, any user who is able to execute applications which grant one or more Linux capabilities to be used in an unconditional manner is considered an authorized administrator. Note: the process executing on behalf of the root user must possess MLS override attributes to perform management aspects of the Mandatory Access Control Policy.

Classification

A sensitivity label associated with an object.

Clearance

A sensitivity label associated with a subject or user.

Data

Arbitrary bit sequences on persistent or transient 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).

Information

Any data held within a server, including data in transit between systems.

Named Object

In Linux, those objects that are covered by access control policies. The list of objects defined as named objects is provided with FDP_ACC.1.

Object

For Linux, objects are defined by FDP_ACC.1.

Product

The term product is used to define software components that comprise the Wind River Linux system.

Sensitivity Label

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 categories. The policy defines the number and names of the sensitivity levels and categories.

Subject

There are two classes of subjects in WRLS: i) untrusted internal subject - this is a Linux process running on behalf of some user or providing an arbitrary service, running outside of the TSF (for example, with no privileges); ii) trusted internal subject - this is a Linux process running as part of the TSF (for example: service daemons and the process implementing the identification and authentication of users).

Target Of Evaluation (TOE)

The TOE is defined as the Wind River 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 relies on hardware / firmware functions to implement this proper separation and isolation mechanisms required by ADV_ARC.1.

User

Any individual/person or technical entity (such as a service added by the administrator on top of the TOE) who has a unique user identifier and who interacts with the Wind River Linux product.

User Security Attributes

Defined by functional requirement FIA_ATD.1, every user is associated with a number of security attributes which allow the TOE to enforce its security functions on this user. This also includes the user clearance which defines the maximum sensitivity label a user can have access to.

8.3 References

CC **Common Criteria for Information Technology Security Evaluation**
Version 3.1R3
Date July 2009
Location http://www.commoncriteriaportal.org/files/ccfiles/CCPART1V3.1R3.pdf
Location http://www.commoncriteriaportal.org/files/ccfiles/CCPART2V3.1R3.pdf
Location http://www.commoncriteriaportal.org/files/ccfiles/CCPART3V3.1R3.pdf

ECG **EAL4 Evaluated Configuration Guide for WindRiver Linux Secure 1.0**
Version 1.0
Date 2011-01-14
File name WRLS1-EAL4-Configuration-Guide.pdf

niap-ospp **US Government Protection Profile for General-Purpose Operating Systems in a Networked environment**
Version 1.0
Date 2010-08-30

WRLSAG **Wind River Linux Secure Administrator's Guide**
Version 1.0
Date 2011-01-14
File name wr_linux_secure_admin_guide_1.0.pdf

WRLSCG **Wind River Linux Secure Configuration Guide**
Version 1.0
Date 2011-01-14
File name wr_linux_secure_config_guide_1.0.pdf