EAL4 Evaluated Configuration Guide for Red Hat Enterprise Linux

July 6, 2012; v1.3

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

This document is derived from the "MLS EAL4 Evaluated Configuration Guide for Red Hat Enterprise Linux on IBM hardware" Copyright (c) 2003, 2004, 2005, 2006, 2007 by atsec GmbH, and IBM Corporation or its wholly owned subsidiaries, and the "Common Criteria EAL4+ Evaluated Configuration Guide for Red Hat Enterprise Linux 5 on HP Hardware"

Contents

Chapter 1

Introduction

1.1 Purpose of this document

The Red Hat Enterprise Linux (RHEL) distribution is designed to provide a secure and reliable operating system for a variety of purposes. Because security requirements obviously depend on the applications and environment, it is not possible to simply certify that the system is "secure", a more precise definition is needed.

The Common Criteria (CC) provides a widely recognized methodology for security certifications. A CC evaluation is fundamentally a two-step process, consisting of defining the "security target" which describes the features that are to be evaluated, and then testing and verifying that the system actually implements these features with a sufficient level of assurance.

This document is a security guide that explains how to set up the evaluated configuration, and provides information to administrators and ordinary users to ensure secure operation of the system. It is intended to be self-contained in addressing the most important issues at a high level, and refers to other existing documentation where more details are needed.

The document primarily addresses administrators, but the section "Security guidelines for users" is intended for ordinary users of the system as well as administrators.

Knowledge of the Common Criteria is not required for readers of this document.

1.2 How to use this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 (http://www.ietf.org/rfc/rfc2119.txt)

Note that this document avoids the terms "SHOULD" and "SHOULD NOT" that are defined in RFC 2119. Requirements are either absolute (and marked with MUST and equivalent terms), or entirely optional (in the sense of not affecting required security functions) and marked with RECOMMENDED, MAY or OPTIONAL.

If you follow the requirements in this document when setting up and using the system, your configuration will match the evaluated configuration. Certain configuration options are marked as OPTIONAL and you MAY modify them as needed, but you MUST NOT make other changes, because they will make the system fail to match the evaluated configuration.

Of course, you MUST always use common sense. This document is not a formal specification, and legitimate reasons can exist to modify the system setup in ways not described here if that is necessary for the system to fulfill its intended purpose. Specifically, applying security patches released by the vendor is strongly RECOMMENDED even though that will cause a deviation from the evaluated configuration.

In cases where the requirements and recommendations in this document conflict with those in other sources (such as the online documentation), the information in this Configuration Guide has higher precedence. You MUST follow the steps described here to reach the evaluated configuration, even if other documentation describes different methods.

The evaluated configuration may be set up in two separate ways:

- In **Base mode**, the system supports the usual discretionary access control features and a single "root" administrative account.
- In **MLS mode**, mandatory access control and role-based access control based on the SELinux MLS policy add restrictions to the discretionary access control.

The information in this guide generally applies to both modes except where it specifically refers to "MLS mode" or "Base mode". All references to roles (other than "root") or multilevel security (MLS) always apply only to MLS mode.

The usual convention is used in this guide when referring to manual pages that are included in the software distribution. For example, the notation *ls*(1) means that running the `man -S 1 ls` command will display the manual page for the *ls* command from section one of the installed documentation. In most cases, the `-S` flag and the section number can be omitted from the command, they are only needed if pages with the same name exist in different sections,

1.3 Requirements and assumptions

1.3.1 What is a CC compliant system?

A system can be considered to be "CC compliant" if it matches an evaluated and certified configuration. This implies various requirements concerning hardware and software, as well as requirements concerning the operating environment, users, and the ongoing operating procedures.

Strictly speaking, an evaluation according to the CC represents the results of investigation of the security properties of the target system according to defined guidelines. It should not be considered as a guarantee for fitness for any specific purpose, but should provide help in deciding the suitability of the system considering how well the intended use fits the described capabilities. It is intended to provide a level of assurance about the security functions that have been examined by a neutral third party.

The software MUST match the evaluated configuration. In the case of an operating system, this also requires that the installed kernel, system, and application software are the same. The documentation (including this guide) will specify permitted variations, such as modifying certain configuration files and settings, and installing software that does not have the capability to affect the security of the system (typically those that do not require root privileges). Please refer to section §4.4 "Installation of additional software" of this guide for more information.

Stated requirements concerning the operating environment MUST be met. Typical requirements include a secure location for the hardware (protected from physical access by unauthorized persons), as well as restrictions concerning permitted network connections.

The operation of the system MUST be in agreement with defined organizational security policies, to ensure that actions by administrators and users do not undermine the system’s security.

1.3.2 Hardware requirements

The hardware MUST be one of the following hardware systems. This entire document applies to all hardware systems unless explicitly noted.

The term "virtualization support" applies to the use of KVM and the support of guest systems on the respective platform.

RHEL with virtualization support

IBM based on x86 64bit Intel Xeon processors

- IBM System x: x3400 M2, x3400 M3, x3500 M2, x3500 M3, x3550 M2, x3550 M3, x3620 M3, x3630 M3, x3650 M2, x3650 M3
- IBM BladeCenter: HS22 and HS22V
- IBM iDataPlex: dx360 M2, dx360 M3
- IBM X5 systems: x3850 X5, x3950 X5, x3690 X5

HP based on x86 64bit Intel Xeon processors

- HP Proliant ML 100 and 300 series G6 product line
- HP Proliant DL 100 series G6, 300 series G6 and G7, 500 series G7, 900 series G7 product line
- HP ProLiant BL 200 series G6 and G7, 400 series G6 and G7, 600 series G5, 600 series G7 product line
- HP ProLiant SL 100 series G6 product line

HP based on AMD Opteron processors

- HP Proliant DL 100 series G7, 500 series G7, 700 series G6 product line
- HP Proliant DL 385 G7
- HP ProLiant BL 600 series G6 product line
- HP ProLiant SL 100 series G7 product line

SGI UV

- SGI UV 1000

Dell based on x86 64bit Intel processors

- Dell PowerEdge R720, R620, R820, R520, R320, R420
- Dell PowerEdge T620, T420, T320
- Dell PowerEdge M620, M520, M420, M820

RHEL except virtualization support

- All machines listed for "RHEL with virtualization support"
- IBM System p based on Power 7 processors
- IBM System z based on z/Architecture processors

These systems must be installed with the 64-bit version of RHEL 6.2.

Running the certified software on other similar hardware might result in an equivalent security level, but the certification does not apply if the hardware is different from that used for the testing processes during the evaluation.

Note, the proper operation of all aspects of the software is only ensured when using the aforementioned hardware systems as several hardware mechanisms which may not be present in other systems are vital for the security of the system.

Please refer to section §2.1 "Supported hardware" for more information about additional hardware supported for use with the evaluated configuration.

1.3.3 Requirements for the system's environment

The security target covers one or more systems running RHEL, networked in a non-hostile network, with a well-managed and non-hostile user community. It is not intended to address the needs of an Internet-connected server, or the case where services are to be provided to potentially hostile users.

It is assumed that the value of the stored assets merits moderately intensive penetration or masquerading attacks. It is also assumed that physical controls in place would alert the system authorities to the physical presence of attackers within the controlled space.

You MUST set up the server (or servers) in a physically secure environment, where they are protected from theft and manipulation by unauthorized persons.

You MUST ensure that all connections to peripheral devices and all network connections are protected against tampering, tapping and other modifications. Using the secured protocol of SSHv2 is considered sufficient protection for network connections. All other connections must remain completely within the physically secure server environment.

When using CIPSO-based labeled networking (configured with the *netlabel* application) in MLS mode, all network connections need to reside within the controlled access facilities because the secured protocol of SSHv2 does not protect the label information. Internal communication paths to access points such as terminals or other systems are assumed to be adequately protected.

1.3.4 Requirements for connectivity

All components in the network such as routers, switches, and hubs that are used for communication are assumed to pass the user data reliably and without modification. Translations on protocols elements (such as NAT) are allowed as long as those modifications do not lead to a situation where information is routed to somebody other than the intended recipient system. Network and peripheral cabling must be approved for the transmittal of the most sensitive data held by the system.

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

Be aware that information passed to another system leaves the control of the sending system, and the protection of this information against unauthorized access needs to be enforced by the receiving system. If an organization wants to implement a consistent security policy covering multiple systems on a network, organizational procedures MUST ensure that all those systems can be trusted and are configured with compatible security configurations enforcing an organization wide security policy. How to do this is beyond the scope of this Configuration Guide. If you set up a communication link to a system outside your control, please keep in mind that you will not be able to enforce any security policy for any information you pass to such a system over the communication link or in other ways (for example, by using removable storage media). In MLS mode, the system supports labeled networking which can help ensure consistent handling of data labels across network connections. You MUST nevertheless ensure that all systems and networks involved are configured securely.

Please be also aware that when configuring KVM guest system, KVM allows physical network adapters to be shared among different guest systems. The sharing is based on IP addresses where KVM assigns a unique IP address to each guest. You should consider the configuration of shared network adapters akin to the use of physical switches in your network. If you want to achieve a stronger separation between the virtual machines, each guest domain should be assigned a dedicated physical network interface.

1.3.5 Requirements for procedures (MLS mode only)

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

Procedures MUST exist for establishing the security level of all information imported into the system, for establishing the security level for all peripheral devices (e.g., printers, tape drives, disk drives) attached to the system, and marking a sensitivity label on all output generated.

1.3.6 Requirements for administrators

There MUST be one or more competent individuals who are assigned to manage the system and the security of the information it contains. These individuals will have sole responsibility for the following functions: (a) create and maintain roles (b) establish and maintain relationships among roles (c) Assignment and Revocation of users to roles. In addition these individuals (as owners of the entire corporate data), along with object owners will have the ability to assign and revoke object access rights to roles.

The system administrative personnel MUST NOT be careless, willfully negligent, or hostile, and MUST follow and abide by the instructions provided by the administrator documentation.

In Base mode, every person that has the ability to perform administrative actions by switching to root has full control over the system and could, either by accident or deliberately, undermine security features of the system and bring it into an insecure state. In MLS mode, the system can restrict actions of root users, but it is still REQUIRED that everyone with administrative access to the system must be a trusted administrator. This Configuration Guide provides the basic guidance how to set up and operate the system securely, but is not intended to be the sole information required for a system administrator to learn how to operate Linux securely.

It is assumed, within this Configuration Guide, that administrators who use this guide have a good knowledge and understanding of operating security principles in general and of Linux administrative commands and configuration options in particular. We strongly advise that an organization that wants to operate the system in the evaluated configuration nevertheless have their administrators trained in operating system security principles and RHEL security functions, properties, and configuration.

Every organization needs to trust their system administrators not to deliberately undermine the security of the system. Although the evaluated configuration includes audit functions that can be used to make users accountable for their actions, an administrator is able to stop the audit subsystem and reconfigure it such that his actions no longer get audited. Well trained and trustworthy administrators are a key element for the secure operation of the system. This Configuration Guide provides the additional information a system administrator should obey when installing, configuring and operating the system in compliance with the requirements defined in the Security Target for the Common Criteria evaluation.

The above stated assumptions imply that the DAC and SELinux (if in MLS mode) permissions of system directories, system binary files and their configuration files are left unchanged. Among others, this ensures that only administrators can add new trusted software into the installation.

To ensure the integrity of the system, you MUST schedule periodical reviews of the system operation and system integrity. For example, an integrity verification using the `rpm` tool may be invoked. Another possibility of validating the integrity of the system is the use of `aide`.

The administrator MUST NOT read the contents of the file holding the seed information in */var/lib/random-seed*. This ensures that if an administrator is relieved from his duties and therefore the trust relationship is lifted, the administrator is not able to obtain or apply (partial) knowledge of the state of */dev/random* or */dev/urandom*.

1.3.7 Requirements for the system's users

The security target addresses the security needs of cooperating users in a benign environment, who will use the system responsibly to fulfill their tasks.

Authorized users possess the necessary authorization to access at least some of the information managed by the system and are expected to act in a cooperating manner in a benign environment.

Note that system availability is *not* addressed in this evaluation, and a malicious user could disable a server through resource exhaustion or similar methods.

The requirements for users specifically include:

- User accounts MUST be assigned only to those users with a need to access the data protected by the system, and who MUST be sufficiently trustworthy not to abuse those privileges. For example, the system cannot prevent data from being intentionally redistributed to unauthorized third parties by an authorized user.
- Rights for users to gain access and perform operations on information are based on their membership in one or more roles. These roles are granted to the users by the administrator. These roles MUST accurately reflect the users job function, responsibilities, qualifications, and/or competencies within the enterprise.
- A limited set of users is given the rights to create new data objects and they become owners for those data objects. The organization is the owner of the rest of the information under the control of system.
- Users are trusted to accomplish some task or group of tasks within a secure IT environment by exercising complete control over their data.
- All users of the system MUST be sufficiently skilled to understand the security implications of their actions, and MUST understand and follow the requirements listed in section §7 "Security guidelines for users" of this guide. Appropriate training MUST be available to ensure this.

It is part of your responsibility as a system administrator to verify that these requirements are met, and to be available to users if they need your help in maintaining the security of their data.

Chapter 2

Installation

The evaluation covers a fresh installation of RHEL Version 6.2 Server, on one of the supported hardware platforms as defined in section §1.3.2 "Hardware requirements" of this guide.

The evaluated configuration MUST be the only operating system installed on the server.

2.1 Supported hardware

You MAY attach the following peripherals without invalidating the evaluation results. Other hardware MUST NOT be installed in or attached to the system.

- Any storage devices and backup devices supported by the operating system (this includes hard disks, CD-ROM drives, floppy disk drives and tape drives).
- All Ethernet network adapters supported by the operating system. Modems, ISDN and other WAN adapters are not part of the evaluated environment.
- PCL 4 or PostScript level 1 compatible printers attached to the system using a parallel port or USB connection. In Base mode only, you MAY also use a network printer. Please refer to section §3.5 "Setting up the Cups printing system" of this guide for more information about printing.
- Operator console consisting of a keyboard, video monitor, and optionally mouse. Additionally, you MAY directly attach supported serial terminals (see section §4.10 "Using serial terminals" of this guide), but *not* modems, ISDN cards, or other remote access terminals.

USB keyboards and mice MAY be attached. If a USB keyboard or mouse is used, it MUST be connected before booting the operating system, and NOT added later to a running system. Other hot-pluggable hardware that depends on the dynamic loading of kernel modules MUST NOT be attached. Examples of such unsupported hardware are USB and IEEE1394/FireWire peripherals other than mice and keyboards.

2.2 Selection of install options and packages

This section describes the detailed steps to be performed when installing the RHEL operating system on the target server.

All settings listed here are REQUIRED unless specifically declared otherwise.

2.2.1 Prerequisites for installation

You will need the following components to install a system in the evaluated configuration as explained in the following sections:

- The target system that will be installed, refer to section §1.3.2 "Hardware requirements" of this guide for the list of supported hardware. The target system REQUIRES at least one local hard drive that will be erased and repartitioned for use by the evaluated configuration.
- A static IP address if you are intending to attach the target system to a network; the evaluated configuration does not support DHCP. In addition, you will need to configure the netmask, gateway, and DNS server list manually.
- An Internet-connected system equipped with the *rpm* and *rpm2cpio* package management tools. This system does not need to be in the evaluated configuration, and no packages will be installed on it. It is used to download and verify the installation packages.
- A method to transfer the kickstart installation configuration and RPM packages to the target system. You can use any *one* of the following choices:
  - A CD-R containing the installation files.
  - A USB memory stick or USB external hard drive formatted using either the *vfat* or *ext2* file system.
  - A network server configured to provide the installation files via the HTTP or NFS protocol.

  Note that a floppy disk drive is not suitable due to insufficient capacity.

2.2.2 Preparing for installation

You MUST download the distribution ISO images from the Red Hat Network on a separate Internet-connected computer, and either burn CD-Rs from them, or make the contents available on a file server via NFS or HTTP. The download location *https://access.redhat.com/downloads/* contains links to the platform-specific images.

You MUST use **Red Hat Enterprise Linux 6.2 Server**. Make sure that you are using the appropriate version for your platform, refer to section §1.3.2 "Hardware requirements" of this guide for the list of supported hardware and the corresponding version needed.

You MUST verify that the SHA256 checksums of the image files are correct. The checksums are shown on the RHN web page, please verify that the web page is encrypted (https:// URL) and has a valid certificate. Then run `sha256sum *.iso` to view the checksums for the downloaded images, and compare them with those shown on the web page.

You MUST download several additional packages not included in the .iso images to set up the evaluated configuration. The packages are available at the Red Hat Network search location *https://rhn.redhat.com/rhn/channels/software/Search.do*.

You MUST use the search mechanism to obtain the following packages:

Common Criteria RPM

- cc-eal4-config-rhel62

RHSA-2012:0124

- kernel 2.6.32-220.4.2.el6
- kernel-firmware 2.6.32-220.4.2.el6
- kernel-headers 2.6.32-220.4.2.el6
- kernel-devel 2.6.32-220.4.2.el6

- kernel-bootwrapper 2.6.32-220.4.2.el6 (PPC64 only)

RHBA-2012:0342

- libvirt 0.9.4-23.el6_2.6 (x86_64 only)
- libvirt-python 0.9.4-23.el6_2.6 (x86_64 only)
- libvirt-client 0.9.4-23.el6_2.6 (x86_64 only)

RHBA-2012:0338

- selinux-policy 3.7.19-126.el6_2.9
- selinux-policy-mls 3.7.19-126.el6_2.9
- selinux-policy-targeted 3.7.19-126.el6_2.9

RHEA-2012:0065

- openssh-clients 5.3p1-70.el6_2.2
- openssh-server 5.3p1-70.el6_2.2
- openssh 5.3p1-70.el6_2.2

RHBA-2012:0134

- policycoreutils 2.0.83-19.21.el6_2
- policycoreutils-newrole 2.0.83-19.21.el6_2
- policycoreutils-python 2.0.83-19.21.el6_2

RHBA-2012:0331

- dracut 004-256.el6_2.1
- dracut-fips 004-256.el6_2.1
- dracut-kernel 004-256.el6_2.1
- dracut-fips-aesni 004-256.el6_2.1 (This RPM is only relevant if you want to use the FIPS 140-2 approved mode **and** your Intel x86 processor provides the AES-NI instruction set – see the FIPS 140-2 security policy for the kernel crypto API for more details. **WARNING:** You MUST NOT install this package if your CPU does not provide the AES-NI instruction set as the kernel would crash during boot.)

RHSA-2012:0050

- qemu-img 0.12.1.2-2.209.el6_2.4 (x86_64 only)
- qemu-kvm 0.12.1.2-2.209.el6_2.4 (x86_64 only)

RHBA-2012:0339

- openswan 2.6.32-10.el6_2

RHBA-2012:0337

- nss-util 3.13.1-3.el6_2
- nspr 4.8.9-3.el6_2

RHBA-2012:0344

- nss 3.13.1-7.el6_2
- nss-sysinit 3.13.1-7.el6_2

RHBA-2012:0360

- openssl 1.0.0-20.el6_2.2 (the automated installation requires that both word sizes, i.e. 32 bit and 64 bit, are installed)
- openssl-devel 1.0.0-20.el6_2.2 (the automated installation requires that both word sizes, i.e. 32 bit and 64 bit, are installed)

RHEA-2012:0486

- libgcrypt 1.4.5-9.el6_2.2 (the automated installation requires that both word sizes, i.e. 32 bit and 64 bit, are installed)

The installation script will prompt for the specific files and version numbers required. Alternatively, search for the variable RPMS_NEEDED in the kickstart file to see the full list of needed packages.

The files needed are the *cc-eal4-config-rhel62* RPM, the unpacked kickstart file (contained within the *cc-eal4-config-rhel62* RPM), and a specific set of RPM packages containing post-RHEL6.2 updates.

Download the RPMs using a separate Internet-connected computer. Do NOT install the downloaded packages yet.

You MUST have the Red Hat package signing key available to verify the integrity of the additional RPM packages. It is available at the following location:

```
https://www.redhat.com/security/650d5882.txt
```

On the download system, run the following commands to verify the package integrity:

```
rpm --import 650d5882.txt
rpm --checksig cc-eal4-config-rhel62-*.rpm
```

This MUST display the status "gpg OK". If it does not, you MUST NOT proceed with the installation using that file.

The web page *https://www.redhat.com/security/team/key/* provides additional information about the usage of package signing keys.

Next, on the download system, unpack the contents of the *cc-eal4-config-rhel62* RPM into a temporary directory:

```
mkdir cc-inst
cd cc-inst
rpm2cpio ../cc-eal4-config-rhel62-*.rpm | cpio -id
```

This will create the following directory structure in the current working directory:

```
# this guide, and supporting documentation
./usr/share/doc/cc-eal4-config-rhel62-*/
        GPL.txt
        RHEL62-EAL4-Configuration-Guide.*

# the kickstart configuration used to automate the installation
./usr/share/cc/kickstart/
        ks-x86_64.cfg          # Xeon, AMD Opteron systems
        ks-ppc64.cfg           # IBM Power systems
        ks-s390x.cfg           # IBM System z
        ks-x86_64+SGI.cfg      # SGI x86 system

# the evaluated configuration reconfiguration script
```

```
./usr/sbin/
        cc-config

# configuration files used for the evaluated configuration
./usr/share/cc/
        auditd.conf
        [...]
        xinetd.conf
```

Depending on the installation method you choose, do *one* of the following steps:

- Burn a CD-R containing the kickstart files from *./usr/share/cc/kickstart/* and the downloaded RPM package(s), with all files at the top directory level (no subdirectories).
- Copy the kickstart files from *./usr/share/cc/kickstart/* and the downloaded RPM packages onto a USB memory stick or USB external hard drive (formatted using either the *vfat* or *ext2* file system). Put the files at the top directory level (no subdirectories).
- Configure a network server to provide the installation files via the HTTP or NFS protocol. Put the downloaded RPM package(s) and the kickstart files from *./usr/share/cc/kickstart/* into a single directory with no subdirectories.

2.2.3 Package selection

The kickstart configuration files contain the package listing which defines the set of packages applied during the initial installation. That package listing consists of a number of mandatory as well as optional packages.

The modification of the package list gives you more freedom to limit the set of installed packages. You MAY disregard the package listing as the default set delivered with the kickstart configuration files installs all packages necessary to cover all functional aspects defined in the Security Target.

If you want to modify the package listing, edit the kickstart configuration file and search for the package listing identified by the string *Package listing*.

Before starting the package listing, the kickstart configuration file explains how to read that listing. It defines different categories of package sets. You MAY remove packages out of the package list based on the definition of the categories and the operational mode intended for the newly installed system.

2.2.4 FIPS 140-2

RHEL provides a central flag to switch various cryptographic libraries as well as cryptographic applications into FIPS 140-2 compliant mode. Although the FIPS 140-2 mode is out of scope for a Common Criteria evaluation, both modes work well together.

If you are not concerned with the FIPS 140-2 compliant operation of cryptographic mechanisms, you MAY skip this section.

The Security Policy documents of the various FIPS 140-2 modules delivered with RHEL can be obtained at the NIST web page at http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140val-all.htm. These documents provide the guidance for bringing the respective module into the FIPS 140-2 compliant mode. All Security Policy modules document how to achieve that mode by configuring RHEL at runtime.

However, RHEL allows the set up of the following FIPS 140-2 modules including the generation of cryptographic keys during the initial installation phase which is not covered in the Security Policy documents:

- OpenSSH sshd server
- dm-crypt partition encryption mechanism

To ensure that the keys are generated using FIPS 140-2 compliant key-generation mechanisms, you MUST use the *fips=1* boot flag outlined in section §2.2.6 "Kickstart".

Using that *fips=1* boot flag ensures the following:

- The kernel is booted with the *fips=1* kernel command line flag as required by the Security Policy documents for the different FIPS 140-2 modules. When using these modules during the initial installation time, all operations are performed compliant to the FIPS 140-2 requirements.
- The kernel command line applied to the newly installed system also contains the *fips=1* as well as the *boot=* flag as outlined in the Security Policy documents for the different FIPS 140-2 modules to ensure the newly installed system is booted in FIPS 140-2 mode.
- The `dracut` boot mechanism contains the `dracut-fips` extension which ensures that the boot-time self-tests as well as integrity tests for the kernel are performed.

Therefore, after booting the installation system with the *fips=1* flag, all system setup configuration steps outlined in the various Security Policy documents are already covered.

Note that boot procedures with *fips=1* may take longer and sometimes much longer than regular boot procedures as the FIPS 140-2 self tests, integrity tests and random number generator seeding procedures must be performed.

For more information about the FIPS 140-2 compliant operation of RHEL, please consult the Security Policy document applicable for the intended cryptographic module.

2.2.5 Customizing the installation

You MAY make changes to specific sections of the kickstart configuration. You MUST NOT change any settings not explicitly listed in this section.

`keyboard`

Default: `us`

You MAY select a different keyboard mapping.

`langsupport`

Default: `-default=en_US.UTF-8 en_US.UTF-8`

You MAY add additional language support, but MUST NOT change the default language or remove the `en_US` language support. (Users MAY configure individual language preferences to override the default.)

`timezone`

Default: `America/Chicago`

You MAY select a different time zone.

`firewall`

Default: `-disabled`

You MAY enable the firewall and modify the firewall settings. Please refer to section §4.14 "Firewall configuration" of this guide for more information.

`selinux`

Default: `-enforcing`

For MLS mode, you MUST leave SELinux in enforcing mode. You MAY disable SELinux in Base mode.

gen_partitioning()

You MAY modify the default partitioning scheme in this function in the kickstart file, search for the following comment text:

```
        ## Required partitions, resize as appropriate
        ## Optional partitions, (de)activate and resize as appropriate
```

Note that you will have an opportunity to modify the partition settings during the install, please refer to section §2.2.8 "Partitioning" of this guide for more information. Alternatively, you MAY use the Logical Volume Manager (LVM) to resize and add partitions after the installation is complete as documented in the *lvm*(8) manual page.

2.2.6 Kickstart

It is RECOMMENDED that you disconnect all network connections until the post-install system configuration is finished. You MAY use a network if required for the installation (for example when using a NFS or HTTP network server instead of CD-ROMs). If you do use a network, you MUST ensure that this network is secure.

Launch the installation boot program contained on the CD-ROM. The details of how to do this depend on the hardware platform, please refer to the hardware manuals and the *Red Hat Enterprise Linux Installation Guide*. Typically, insert the first CD and boot from CD-ROM. When obtaining the installation guide via the Internet, you should use an SSL-protected HTTP connection to ensure the integrity and authenticity of the documentation, like *https://www.redhat.com/docs/manuals/enterprise/#RHEL6*.

At the boot loader prompt, you MUST initiate the preconfigured "kickstart" install using a configuration file specific for the evaluated configuration. The installer supports multiple methods to locate the kickstart information file.

You MAY use DHCP to temporarily configure the network during the installation process, but you MUST assign a static IP address for use in the evaluated configuration.

Please refer to the *Red Hat Enterprise Linux Installation Guide* for more information.

You MUST use the `ks=` boot parameter that selects a kickstart based automated installation.

Choose the appropriate kickstart file for your architecture and distribution:

```
    ks-x86_64.cfg
    ks-ppc64.cfg
    ks-s390x.cfg
    ks-x86_64+SGI.cfg
```

The installation process will prompt for all needed information, alternatively you MAY supply the following command line parameters to automate the installation:

`method`

Select one of the supported methods for accessing the distribution media:

```
        method=cdrom:
        method=nfs:server.example.com:/path/to/files/
        method=http://server.example.com/path/to/files/
        method=hd://sda1/path/to/files/
```

ksdevice

Use this network interface for the kickstart installation, default `eth0`.

ip, netmask, gateway, dns

Configure the network parameters for the installation. See also `ksdevice`.

hostname

Specify the fully qualified host name for the system, for example:

```
hostname=rhel6.example.com
```

instdisk

Delete all data from the specified disk and partition it for the evaluated configuration. This will **DESTROY** the data on this disk without prompting, use with care. Example:

```
instdisk=sda
```

console

You MAY use a serial console to control the installation.

You MAY use a computer using terminal emulation software and a null modem cable instead of a standalone serial terminal. You MUST ensure that the serial terminal is secure.

fips

You MAY enable the FIPS 140-2 compliant mode by using the following boot flag

```
fips=1
```

Examples:

```
# kickstart on USB storage device, install from CD
ks=hd:sda1:/ks-x86_64.cfg method=cdrom:


# interactive network install, get IP address via DHCP
ks=http://example.com/rhel/ks-x86_64.cfg


# noninteractive network install (all on a single line)
ip=172.16.2.5 netmask=255.255.255.0 gateway=172.16.2.1
      dns=172.16.2.1
      ks=http://example.com/rhel/ks-x86_64.cfg
      method=cdrom:
      hostname=rhel6.example.com
      instdisk=sda


# kickstart on USB storage device, install from CD, FIPS 140-2 mode
ks=hd:sda1:/ks-x86_64.cfg method=cdrom: fips=1
```

2.2.7 Pre-install configuration

The following transcript shows an example of the interactions during the pre-install phase of the configuration:

```
-------------------------------------------------------------------------------
*** Common Criteria configuration kickstart ***


Using volume group 'VolGroup01'.
(Answer '!' at any prompt to get an interactive shell)


Installation source [cdrom:] ?


Available destination disks:
sda 3067.09716797


Install on which disk(s), comma separated [sda] ?


Disk encryption uses LUKS key setup with
FIPS 140-2 compliant 256 bit AES-XTS plain64 IV with SHA-1
Encrypt all partitions (except /boot) [n] ? y


Hostname (fully qualified) [rhel.example.com] ?


Network interface [eth0] ?


IP address [] ? 172.16.2.5


Netmask [255.255.255.0] ?


Gateway [] ? 172.16.2.1


Nameserver list (comma separated) [] ?


Manually edit partitioning instructions (y/n) [n] ?


--- WARNING -------------------------------------------------
This is your last chance to stop the installation. Continuing
will erase the destination disk and install non-interactively.
Answer 'n' if you need to edit your settings.


Okay to proceed with install on sda (y/n) [n] ? y
-------------------------------------------------------------------------------
```

In case the installation does not show the pre-install configuration prompts, for example if you see a blank screen only, try using a different terminal emulator to control the installation.

2.2.8 Partitioning

You MAY manually edit the partitioning instructions during the kickstart process. This section describes the partitioning requirements.

Set up the REQUIRED / (root) and */var/log* partitions, and as many additional mounted partitions as appropriate. */var/log* REQUIRES at least 100 MB of space in order to be able to install and launch the audit system, but this does not include the additional space needed for saved audit logs. You MAY use a */var/log/audit/* partition separate from */var/log/* to ensure that audit data is stored separately from other system logs. Please refer to section §6.3 "Configuring the audit subsystem" of this guide for more information.

Some configurations (recognized automatically by the installation program) need a separate */boot* partition formatted as an **ext3** or **ext4** file system. If the installation program warns about the partitioning being invalid and that it may result in an unbootable system, add the */boot* partition.

It is RECOMMENDED to also use separate partitions for */var*, */var/log/audit/*, */home* and */tmp*. The following table shows a RECOMMENDED partitioning scheme together with minimum sizes for the partitions. Using more space is RECOMMENDED:

```
/boot           200 MB # if needed by installer
/              2000 MB
/tmp            200 MB
/home           100 MB
/var            500 MB
/var/log        500 MB
/var/log/audit  100 MB # needed for install, >>1GB for use
```

All mounted partitions MUST be of type **ext3**, **ext4** or **swap** and **formatted**.

If you want to use disk encryption, please refer to section §2.2.9 "Hard disk encryption".

In MLS mode, the polyinstantiation mechanism changes the location that file data is stored. The automated installation process ensures that a separate partition for */tmp* will be mounted as */tmp-parent* which hosts all temporary file systems. Subdirectories in */tmp-parent* will be created and automatically mounted using bind mounts to cover the locations of */tmp*, */var/tmp* and the polyinstantiated */tmp* directories for users. Please refer to section §3.8 "Configuring polyinstantiation" of this guide for more information.

Furthermore, when using the KVM functionality, the default location for file-based disk images for virtual machines are stored in */var/lib/libvirt/images* per default. As disk images can potentially be very large, a separate partition for hosting these images MAY be defined. Note that `libvirtd` allows the configuration of alternative locations for these disk images.

Configuring a swap partition at least as large as the installed RAM is RECOMMENDED.

2.2.9 Hard disk encryption

During the initial installation phase you MAY encrypt the hard disk using the dm-crypt partition encryption mechanism. The installation process allows you to encrypt either the entire hard disk (except */boot*) or partitions chosen by you.

If you chose the option of full disk encryption outlined in section §2.2.7 "Pre-install configuration", the installer asks you to provide a number of key strokes. These key strokes are vital to ensure the system has enough entropy to produce a strong cryptographic key. As advertised by the installer you MUST NOT simply press one key and hold it down, but randomly typing some keys.

The following hints about disk encryption are only valid if you chose a manual partitioning as outlined in section §2.2.8 "Partitioning". If you chose the option of full disk encryption outlined in section §2.2.7 "Pre-install configuration", you MAY skip this section.

You MUST NOT encrypt */boot* as this would impair the ability to boot the system. You MUST configure a separate */boot* partition if you intend to configure partition encryption for the root partition of */*.

When you manually edit the partitioning schema, you MAY append the option *–encrypt* to the respective partition definition. In this case, the partition will be encrypted using dm-crypt. If you apply *–encrypt* to more than one partition, all such marked partitions will be encrypted using the same cryptographic key protected with the same passphrase. The installer will ask you for a passphrase that will be used to protect the key.

Before you close the editor for manual partitioning, you MUST ensure that you typed at least 256 key strokes to ensure sufficient entropy is present for generating a strong key. **WARNING:** You MUST perform this operation on the local console as the operation is intended to cause interrupts which fill the entropy pool of the Linux kernel. Only with sufficient entropy, the master key for the disk encryption schema is sufficiently strong. If you access the installation system remotely, e.g. via serial console or via network, you must log in locally at this point and type in 256 key strokes. The key strokes on the remote system do not update the entropy pool on the local system.

2.2.10 Post-install configuration

The system will run the *cc-config* script to automatically configure the initial system settings, then prompt to reboot.

The following transcript shows an example of the interactions during the post-install phase of the configuration (the exact version numbers and package lists may differ in the final version). Note that this transcript shows that the administrator performs an NFS mount to access the RPM packages from an NFS server instead of fetching them from a web server.

```
--------------------------------------------------------------------------------
*** Common Criteria configuration kickstart ***

Operational mode (base or mls) [base] ? mls

Please verify the system time and date:
    Local time:           Wed Apr 25 00:28:16 CDT 2011
    Universal time (UTC): Wed Apr 25 05:28:16 UTC 2011

If the time or time zone is wrong, please correct it now using
tools such as 'date', 'hwclock', or 'tzselect' as appropriate.

Is the time correct (y/n) [y] ?
Bringing up loopback interface:  [  OK  ]
Bringing up interface eth0:  [  OK  ]

Need to install the certification RPM and updated RPM packages:

cc-eal4-config-rhel62-0.10-1.noarch.rpm
[...]

Supply a web URL or a local (absolute) directory name.

If you need to mount a device containing the files,
enter '!' and RETURN to get a shell prompt.

Location [http://local.install.web.server/RHEL62/addtl_RPMs/] ? !
```

```
Starting interactive shell, type 'exit' when done
sh-3.1# mount -o nolock 172.16.2.1:/home/export /mnt
sh-3.1# exit
exit

Location [http://local.install.web.server/RHEL62/addtl_RPMs/] ? /mnt/rpms/
`/mnt/rpms/acl-2.2.39-2.1.el5.i386.rpm' -> `./acl-2.2.39-2.1.el5.i386.rpm'
[...]
`/mnt/rpms//vixie-cron-4.1-68.el5.i386.rpm' -> `./vixie-cron-4.1-68.el5.i386.rpm'

Preparing...                ########################################### [100%]
   1:audit-libs             ########################################### [  3%]
[...]
  31:vixie-cron             ########################################### [100%]

Switching SELinux to MLS mode...
Fixing file labels...
/sbin/setfiles:  labeling files under /
***************************************************

Please enter the password for the root account.
Changing password for user root.
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully.

Create an administrative user account.

Real name (First Last) [] ? John Doe

Userid [jdoe] ?
Changing password for user jdoe.
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully.

Add more administrative users (y/n) [n] ?
 --- Wed Apr 25 00:34:47 CDT 2009 script running: /usr/sbin/cc-config args -a --base
###  Configure mount options in /etc/fstab
[...]
### System reboot

Reconfiguration successful.
It is now necessary to reboot the system.
After the reboot, your system configuration will match the evaluated configuration
*** Reboot the system? (y/n) [y]: y
rebooting the system now. Sleeping for 10 seconds...
+sync
+sleep 10
Exiting.
-------------------------------------------------------------------------------
```

Warning messages indicating duplicate configuration files at this stage are harmless and can be ignored, for example:

```
  warning: /etc/pam.d/system-auth created as /etc/pam.d/system-auth.rpmnew
```

The output of the *cc-config* script is stored in the */var/log/cc-config.log* file.

2.3 Architecture specific manual configuration steps

The following sections provide manual configuration steps required for specific hardware systems. If you install the system on one of the mentioned hardware, you MUST perform the outlined steps.

2.3.1 SGI UV

The SGI UV system requires additional kernel command line flags. These flags have to be added to the kernel command line specified in the boot loader configuration file */boot/efi/EFI/redhat/grub.conf*.

The boot loader configuration file contains the kernel command line in the line started with "kernel". For example, the following shortened line may be present

```
 kernel /vmlinuz-2.6.32-220.2.1.el6.x86_64 ro root=/dev/mapper/VolGroup01-root...
```

You MUST append the following parameters on that line. Note, all parameters MUST be added on the same line.

```
        add_efi_memmap
        log_buf_len=8M
        nmi_watchdog=0
        nohz=off
        nortsched
        processor.max_cstate=1
        earlyprintk=ttyS0,115200n8
```

In addition, you MUST append the following option. In case the option is already present in the kernel command line but just with a different value, you MUST replace the existing value with the new value.

```
        crashkernel=512M
```

Chapter 3

Secure initial system configuration

After the initial installation using the procedure described in the previous chapter, the operating system is in the evaluated configuration if you have selected a Base mode installation.

In MLS mode, you MUST appropriately configure CIPSO or IPSEC labeled networking as described in section §4.18 "Labeled networking (MLS mode only)" of this guide, as there is no universally applicable default for this. Your system will be in the evaluated configuration once labeled networking is activated.

The system does not define audit rules as there is no universally applicable default for this. Please refer to section §6.3 "Configuring the audit subsystem" of this guide for more information.

The steps described in this chapter were done automatically if the kickstart install has completed successfully. You MAY skip ahead to section §4 "System operation" of this guide.

The information in this section provides background information about how this configuration was achieved, and mentions some changes you MAY make to the installed system while still remaining within the evaluated configuration. It is not intended to be a complete listing of the changes made to the system. Following the instructions in section §2 "Installation" of this guide is the only supported method to set up the evaluated configuration.

After software upgrades or installation of additional packages, you MUST ensure that the configuration remains secure. Please refer to sections §1.2 "How to use this document" and §4.4 "Installation of additional software" of this guide for additional information. You MAY re-run the *cc-config* script, but this does not guarantee that you will be in the evaluated configuration if you have added, deleted, modified, or replaced system components.

3.1 Creating additional user accounts for administrators

The evaluated configuration disables direct root login over the network. All system administrators MUST log in using a non-root individual user ID, then use the *su*(8) command to gain superuser privileges for administrative tasks. This requires membership in the 'wheel' group of trusted users.

You MUST define at least one non-root user account with the *useradd*(8) command, and add this user account to the 'wheel' group. Note that the enhanced password quality checking mechanisms and the password expiry settings of the evaluated configuration are not active yet. You must manually set the password properties in accordance with the password policy.

Please refer to section §4.9.7 "Defining administrative accounts" of this guide for more information about creating administrative accounts. The administrative accounts created during the initial install (§2.2.10 "Post-install configuration") are *staff_u* users as described in that section.

Please refer to sections §4.9 "Managing user accounts" and §7.3 "Password policy" of this guide for more information on creating user accounts.

3.2 Installing required updates

Several packages shipped on the installation media MUST be replaced with more recent versions to fix bugs or add additional features required for the evaluated configuration.

The kickstart script automatically installs the required updates in the postinstall section.

3.3 Automated configuration of the system

The kickstart script installs the `cc-eal4-config-rhel62` RPM package and runs the *cc-config* script contained within that RPM package non-interactively.

You MAY run the *cc-config* script interactively after installation is complete to verify and reset configuration settings to appropriate values for the evaluated configuration.

The *cc-eal4-config-rhel62* package contains configuration files and the script *cc-config* that sets up the evaluated configuration.

Run the following command to view a summary of the supported options:

```
cc-config -h
```

You MAY use the `-a` flag to automate the install and have it run without prompting. This is intended for people who are familiar with the process; if running it for the first time you SHOULD let it run interactively and verify the actions as described in this guide.

You MUST answer all questions asked by the script that are not marked as "optional" with `y` to achieve the evaluated configuration.

WARNING: The *cc-config* script will reboot the system as the final step in the process. Remember to remove any CD-ROM from the drive and/or configure the system to boot from hard disk only.

3.4 Setting up xinetd

3.4.1 xinetd in MLS mode

In MLS mode, the *xinetd* super server is used in the evaluated configuration to integrate *sshd* and labeled networking.

Please refer to sections §4.18 "Labeled networking (MLS mode only)" and §7.5.2 "Multilevel mandatory access control (MLS mode only)" of this guide for more information about using *ssh* in MLS mode.

3.4.2 xinetd in Base mode

In Base mode, *xinetd* is not used in the evaluated configuration, but MAY be used to start non-root network processes. The file */etc/xinetd.conf* contains default settings, these can be overridden by service-specific entry files stored in the directory */etc/xinetd.d/*.

3.5 Setting up the Cups printing system

Use of the Cups printing system is OPTIONAL, if the service is active you MUST configure the settings described in this section.

You MAY attach a PCL 4 or PostScript level 1 compatible printer to the system using a parallel port or USB connection.

In Base mode only, you MAY also use a network printer. Network printers MUST NOT be used in MLS mode since they cannot meet the requirements for data export.

Verify that the printer daemon is able to access your printer devices with the configured permissions. You MAY need to reconfigure the printer device access rights to match, for example by setting the device owner for the */dev/lp** devices to the *lp* user in the */etc/udev/permissions.d/50-udev.permissions* file.

In MLS mode, you MUST use the *chcon -l* command to assign appropriate MLS levels to the printer device. This MUST be done while the printer queue is disabled. For example:

```
cupsdisable
chcon -t printer_device_t -l SystemLow-SystemHigh /dev/lp1
cupsenable
```

In MLS mode, you MUST ensure that the printer cannot be accessed directly by non-administrative users who try to bypass the operating system print queue. For parallel port and USB connections, the DAC restrictions on the printer device will restrict access to administrators which is sufficient protection.

In MLS mode, note that the printer name is visible for all users, even for users who do not have sufficient clearance to use the printer. The human readable printer name MUST NOT in itself contain sensitive information.

Please refer to the *cupsd.conf*(5), *cupsdisable*(8), *cupsenable*(8), *chcon*(8) and *cupsd*(8) man pages for more information.

3.6 Setting up Postfix (Base mode only)

Postfix is NOT supported in MLS mode. You do not need to specifically disable it but it is unlikely to work as expected since it does not contain any support for multilevel security or polyinstantiation.

Use of the Postfix mail transport is OPTIONAL, but it is strongly suggested to configure a mail transport agent (MTA) to receive cron job reports. if the service is active you MUST configure the settings described in this section.

An alias MUST be set up for root in */etc/aliases*, as postfix will not deliver mail while running with UID 0. Specify one or more user names of administrators to whom mail addressed to root will be forwarded, for example with this entry in the */etc/aliases* file:

```
root: jdoe, jsmith
```

You MUST disable the execution of programs in the *$HOME/.forward* files of individual users. Add the following line to the */etc/postfix/main.cf* file:

```
allow_mail_to_commands = alias
```

Please see *postfix*(1), *master*(8), *local*(8), and the documentation in */usr/share/doc/postfix*/* for details.

3.7 Setting up Cron (MLS mode only)

In MLS mode, Cron MUST be configured to disable sending mail. Edit the */etc/sysconfig/crond* file to use the following setting:

```
CRONDARGS="-m /bin/true"
```

All output from commands run via the cron system will be silently discarded.

Please refer to section §4.5 "Scheduling processes using cron" of this guide for more information.

3.8 Configuring polyinstantiation

The *pam_namespace.so* module ensures that user sessions running at multiple MLS levels each have their private copy of the home directory and temporary directories. The module is configured using the */etc/security/namespace.conf* file.

You MUST retain the polyinstantiated directory configuration provided with the initial installation as these directories are intended to hold data from users with different labels.

You MAY modify the polyinstantiation parent directories (second column) or add additional lines as appropriate. Please refer to the *pam_namespace*(8) and *namespace.conf*(5) man pages for more information. To comply with the MLS requirements, all directories that are writable by unprivileged users belonging to different labels MUST be polyinstantiated.

3.9 Configuring the boot loader

You MUST set up the server in a secure location where it is protected from unauthorized access. Even though that is sufficient to protect the boot process, it is RECOMMENDED to configure the following additional protection mechanisms:

- Ensure that the installed system boots exclusively from the disk partition containing RHEL, and not from floppy disks, USB drives, CD-ROMs, network adapters, or other devices.
- Ensure that this setting cannot be modified, for example by using a BootProm/BIOS password to protect access to the configuration.

Chapter 4

System operation

To ensure that the systems remains in a secure state, special care MUST be taken during system operation.

4.1 System startup, shutdown and crash recovery

Use the *shutdown*(8), *halt*(8), *poweroff*(8), or *reboot*(8) programs as needed to shut down or reboot the system.

When powered on (or on initial program load of the logical partition on a host system), the system will boot into the RHEL operating system. If necessary (for example after a crash), a filesystem check will be performed automatically. In rare cases manual intervention is necessary, please refer to the *e2fsck*(8) and *debugfs*(8) documentation for details in this case.

In case a nonstandard boot process is needed (such as booting from floppy disk or CD-ROM to replace a defective hard drive), interaction with the boot loader and/or the host's management system can be used to modify the boot procedure for recovery.

For example, you can use the following grub commands to launch a shell directly from the kernel, bypassing the normal init/login mechanism:

```
# view the current grub configuration
grub> cat (hd0,1)/boot/grub/menu.lst

# manually enter the modified settings
grub> kernel (hd0,1)/boot/vmlinuz root=/dev/sda1 init=/bin/sh
grub> initrd (hd0,1)/boot/initrd
grub> boot
```

Please refer to the relevant documentation of the boot loader, as well as the RHEL administrator guide, for more information.

4.2 Backup and restore

Whenever you make changes to security-critical files, you MAY need to be able to track the changes made and revert to previous versions, but this is not required for compliance with the evaluated configuration.

The *tar*(1) archiver is RECOMMENDED for backups of complete directory contents, please refer to section §7.6 "Data import / export" of this guide. Regular backups of the following files and directories (on removable media such as tapes or CD-R, or on a separate host) are RECOMMENDED:

```
/etc/
/var/spool/cron/
```

You MUST use the `-acls` option for *tar* if you intend to save or restore ACLs. Similarly, you MUST use the `-selinux` option for *tar* if you want to save or restore SELinux labels holding the MLS labels. You MAY omit these options if you only intend to save or restore file contents without security metadata.

Depending on your site's audit requirements, also include the contents of */var/log/* in the backup plan. In that case, the automatic daily log file rotation needs to be disabled or synchronized with the backup mechanism, refer to sections §6.2 "System logging and accounting" and §6.3 "Configuring the audit subsystem" of this guide for more information.

You MUST protect the backup media from unauthorized access, because the copied data does not have the access control mechanisms of the original file system. Among other critical data, it contains the secret keys used by the *SSH* server, as well as the */etc/shadow* password database. Store the backup media at least as securely as the server itself.

A RECOMMENDED method to track changes is to use a version control system. RCS is easy to set up because it does not require setting up a central repository for the changes, and you can use shell scripting to automate the change tracking. RCS is not included in the evaluated configuration, see *rcsintro*(1) in the rcs RPM package for more information. Alternatively, you can create manually create backup copies of the files and/or copy them to other servers using *scp*(1).

4.3 Gaining administrative access

System administration tasks require superuser (root) privileges. In Base mode, superuser rights are also sufficient for administrative actions. In MLS mode, superuser rights are a prerequisite for administrative rights, but in addition you need to select an administrative role with appropriate privileges.

Directly logging on over the network as user root is disabled. To gain superuser rights, you MUST first authenticate using an unprivileged user ID, and then use either the `su` or the `sudo` command to switch identities. Note that you MUST NOT use the root rights for anything other than those administrative tasks that require these privileges, all other tasks MUST be done using your normal (non-root) user ID.

4.3.1 Using su

The `su` command allows a permanent switch of the user ID for the current session.

You MUST use exactly the following *su*(1) command line to gain superuser access:

```
/bin/su -
```

This ensures that the correct binary is executed irrespective of PATH settings or shell aliases, and that the root shell starts with a clean environment not contaminated with the starting user's settings. This is necessary because the *.profile* shell configuration and other similar files are writable for the unprivileged ID, which would allow an attacker to easily elevate privileges to root if able to subvert these settings.

Administrators MUST NOT add any directory to the root user's PATH that are writable for anyone other than root, and similarly MUST NOT use or execute any scripts, binaries or configuration files that are writable for anyone other than root, or where any containing directory is writable for a user other than root.

4.3.2 Using sudo

The `sudo` command allows invoking of a command with a configured user ID, including the root user ID. The switch to the target user ID only remains for the duration of the execution time of the specified command.

The default configuration of `sudo` does not allow any unprivileged users to invoke privileged commands. Depending on your requirements, the following examples may be used as a guide to configure `sudo`. More information may be obtained from the *sudoers*(5) man page.

The following configuration allows all users associated with the *wheel* group to use all commands with privileges:

```
%wheel        ALL=(ALL)       ALL
```

The `sudo` configuration allows specification of the target SELinux type and role the application will be invoked with. The following example defines that *userA* may invoke all commands with the root user ID and the sysadm_r/sysadm_t SELinux role/type:

```
userA ALL=(ALL) ROLE=sysadm_r TYPE=sysadm_t ALL
```

The last example allows a user to only invoke one specific command. With this configuration, delegation of specific duties may be configured. Note, `sudo` even allows specification of command line options that may be used together with the given command.

```
userA ALL=(ALL) ROLE=sysadm_r TYPE=sysadm_t /usr/sbin/useradd
```

4.3.3 Administrative users in MLS mode

In MLS mode, the system supports several administrative roles:

system_r

The operating system supports multiple roles for noninteractive system processes such as daemons. All non-interactive roles are considered to be subdivisions of a conceptual "system" role. The additional restrictions enforced on system services are beyond the scope of this document. The definition of system roles allows separating those from users.

sysadm_r

This is a role defined for general system administration tasks, including setting or modifying security contexts, and changing the sensitivity label of a subject or object.

secadm_r

This is a role defined for administration of security-related configuration items. For example, the configuration of SELinux, including the re-loading of the SELinux policy is only possible using the secadm_r role.

auditadm_r

This is a role for the management of the audit configuration and evaluation of the audit records.

In addition, the system provides the following non-administrative roles by default:

staff_r

This is a role for users that are allowed use the newrole command to transition to administrative roles.

user_r

This is a generic role for all users other than "staff".

In MLS mode, you MUST select one of the administrative roles after running "su" to perform administrative actions, for example:

```
        /bin/su -
        newrole -r sysadm_r
        newrole -r auditadm_r
```

You MUST use the *SystemLow* level for system administration. This is necessary to avoid accidentally upgrading system files to inappropriately high MLS levels, which would make them unreadable for processes running at lower levels.

Please refer to the *newrole*(1) man page and section §7.5.3 "Role-based access control (MLS mode only)" of this guide for more information.

In MLS mode, you MAY also select a MLS level for administrative actions, usually one of *SystemLow* or *SystemHigh*. Please refer to section §7.5.2 "Multilevel mandatory access control (MLS mode only)" of this guide for more information.

4.4 Installation of additional software

Additional software packages MAY be installed as needed, provided that they do not conflict with the security requirements.

4.4.1 Supported software architectures

You MUST use the default kernel (which is SMP capable even on uniprocessor systems) from the package *kernel-2.6.*.rpm* on all systems. You MUST NOT use a different kernel flavor such as the PAE kernel.

You MUST select the appropriate RPM packages for your architecture. The 64bit architectures support execution of both 64bit and 32bit binaries.

x86_64 (Intel EM64T, AMD Opteron)

These systems use a 64bit kernel and 64bit userspace programs and also supports running 32bit programs. Use the ***.x86_64.rpm** or ***.noarch.rpm** variants of packages. You can OPTIONALLY install the ***.i386.rpm** or ***.i686.rpm** variants of libraries (package names containing *-libs* or *-devel*) in addition to the 64bit versions.

System p (ppc/ppc64)

These systems use a 64bit kernel, but the installed userspace programs are the 32bit variants. They support running 64bit programs as well. Use the *.ppc64.rpm kernel for System p. Use the *.ppc.rpm or *.noarch.rpm packages for all packages other than the kernel. You can OPTIONALLY install the *.ppc64.rpm variants of libraries (package names containing -libs or -devel) in addition to the 32bit versions.

System z (s390x)

The evaluated configuration uses a 64bit kernel running 64bit userspace programs. Use the *.s390x.rpm or *.noarch.rpm variants of packages. You can OPTIONALLY install the 32bit *.s390.rpm variants of libraries (package names containing -libs or -devel) in addition to the 64bit versions.

4.4.2 Security requirements for additional software

Any additional software added is not intended to be used with superuser privileges. The administrator MUST use only those programs that are part of the original evaluated configuration for administration tasks, except if the administrator has independently ensured that use of the additional software is not a security risk.

Administrators MAY add scripts to automate tasks as long as those only depend on and run programs that are part of the evaluated configuration.

The security requirements for additional software are:

- Kernel modules other than those provided as part of the evaluated configuration MUST NOT be installed or loaded. You MUST NOT load the *tux* kernel module (the in-kernel web server is not supported). You MUST NOT add support for non-ELF binary formats or foreign binary format emulation that circumvents system call auditing. You MUST NOT activate knfsd or export NFS file systems.
- Device special nodes MUST NOT be manually added to the system.
- SUID root or SGID root programs MUST NOT be added to the system. Programs which use the SUID or SGID bits to run with identities other than root MAY be added if the numerical SUID and SGID values are not less than 500 as defined with the values *UID_MIN* and *GID_MIN* in the configuration file of */etc/login.defs*. This restriction is necessary to avoid conflict with system user and group IDs such as the "disk" group.
- The content, permissions, and ownership of all existing filesystem objects (including directories and device nodes) that are part of the evaluated configuration MUST NOT be modified. Files and directories MAY be added to existing directories provided that this does not violate any other requirement.
- The rules for the `udev` framework provided in */lib/udev/rules.d* must be left unchanged. Also, no new rules must be added to the `udev` framework.
- Programs automatically launched with root privileges MUST NOT be added to the system. Exception: processes that *immediately* and *permanently* switch to a non-privileged identity on launch are permitted, for example by using `su USERID -c LAUNCH_COMMAND` in the startup file, or alternatively by using the *setgroups*(2), *setgid*(2) and *setuid*(2) system calls in a binary. (*seteuid*(2) etc. are insufficient.)

  Automatic launch mechanisms are:

  - Entries in */etc/inittab*
  - Executable files or links in */etc/rc.d/init.d/* and its subdirectories
  - Entries in */etc/xinetd.conf*
  - Scheduled jobs using `cron` (including entries in */etc/cron** files)
  - Applications started using the system DBUS which is configured via */etc/dbus-1/system.d/*.
  - Applications specified in */etc/sudoers* or with rules located in a file in the directory */etc/sudoers.d*. Note, that file may contain the keyword *ALL* as a placeholder for a command. In this case, the user allowed to execute all commands with that rule using the root user ID MUST ensure that additional applications are not executed using `sudo`. This requirement can only be met with operational procedures.
  - Applications spawned via `udev` where the rules are added to */lib/udev/rules.d*.

Examples of programs that usually do not conflict with these requirements and MAY be installed are compilers, interpreters, network services running with non-root rights, and similar programs. The requirements listed above MUST be verified in each specific case.

Some system programs are configured to automatically change their SELinux context when executed. This uses the type transitioning facilities of the SELinux policy, and can add or remove privileges from programs. Type transitioning programs can be recognized by the file context (as shown with the `ls -Z` command) containing the *exec_t* suffix, for example */bin/passwd* with the *passwd_exec_t* type. You MUST NOT assign type transitions with predefined system types to additional programs. Automatic type transitions are not safe for use with script files, and MUST NOT be used for adding privileges to scripts, only for voluntarily removing them.

4.5 Scheduling processes using cron

The *cron* facility is available for scheduling processes. The legacy *at* service is not supported in the evaluated configuration.

The *cron*(8) program schedules programs for execution at regular intervals. Entries can be modified using the *crontab*(1) program - the file format is documented in the *crontab*(5) manual page.

In MLS mode, users MAY use the MLS_LEVEL environment variable to select an MLS level for the job to execute. This is documented in the *crontab*(5) manual page.

In MLS mode, cron will NOT send mail containing job output to users. You MAY use output redirection in crontab entries to save output in files at the appropriate MLS level.

You MUST follow the rules specified for installation of additional programs for all entries that will be executed by the root user. Use non-root crontab entries in all cases where root privileges are not absolutely necessary.

Errors in the non interactive jobs executed by `cron` are reported in the system log files in */var/log/*, and, in Base mode, additionally via e-mail to the user who scheduled it.

Permission for users to schedule jobs with `cron` through the following *allow* and *deny* files:

```
/etc/cron.allow
/etc/cron.deny
```

The *allow* file has precedence if it exists, then only those users whose usernames are listed in it are permitted to use the service. If it does not exist, the *deny* file is used instead and all users who are *not* listed in that file can use the service. Note that the contents of these files are only relevant when the scheduling commands are executed, and changes have no effect on already scheduled commands.

In the RHEL distribution, the *allow* files do not exist, and *deny* files are used to prevent system-internal IDs and/or guest users from using these services. By default, the evaluated configuration permits everybody to use *cron*.

It is RECOMMENDED to restrict the use of *cron* to human users and disallow system accounts from using these mechanisms. For example, the following commands add all system accounts other than root to the *deny* files:

```
awk -F: '{if ($3>0 && $3<100) print $1}' /etc/passwd >/etc/cron.deny
chmod 600 /etc/cron.deny
```

Administrators MAY schedule jobs that will be run with the privileges of a specified user by editing the file */etc/crontab* with an appropriate username in the sixth field. Entries in */etc/crontab* are not restricted by the contents of the *allow* and *deny* files.

You MAY create a */etc/cron.allow* file to explicitly list users who are permitted to use this service. If you do create this file, it MUST be owned by the user root and have file permissions 0600 (no access for group or others).

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 or at 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.

4.6 Mounting filesystems

If any filesystems need to be mounted in addition to those set up at installation time, appropriate mount options MUST be used to ensure that mounting the filesystem does not introduce capabilities that could violate the security policy.

The special-purpose *proc*, *sysfs*, *devpts*, *selinuxfs*, *binfmt_misc*, *devtmpfs*, *mqueue* and *tmpfs* filesystems are part of the evaluated configuration. These are virtual filesystems with no underlying physical storage, and represent data structures in kernel memory. Access to contents in these special filesystems is protected by the normal discretionary access control policy and additional permission checks.

Note that changing ownership or permissions of virtual files and directories is generally NOT supported for the *proc* and *sysfs* filesystems (corresponding to directories */proc/* and */sys/*), and attempts to do so will be ignored or result in error messages.

A new filesystem can be integrated as part of the evaluated configuration, for example by installing an additional hard disk, under the following conditions:

- The device is protected against theft or manipulation in the same way as the server itself, for example by being installed inside the server.
- One or more new, empty, file systems with the file system formats listed in section §2.2.8 "Partitioning" are created on it.
- The file systems are mounted using the `acl` option, for example with the following setting in the */etc/fstab* file:

  ```
  /dev/sdc1 /home2 ext3 acl 1 2
  ```

  Existing files and directories MAY then be moved onto the new file systems.
- If a device containing a file system is ever removed from the system, the device MUST be stored within the secure server facility, or alternatively MUST be destroyed in a way that the data on it is reliably erased.

Alternatively, media MAY be accessed without integrating them into the evaluated configuration, for example CD-ROMs or DVDs.

CD/DVD devices MUST be accessed using the *iso9660* filesystem type. Using an automounter is NOT permitted in the evaluated configuration.

The following mount options MUST be used if the filesystems contain data that is not part of the evaluated configuration:

```
nodev,nosuid
```

Adding the *noexec* mount option to avoid accidental execution of files or scripts on additional mounted filesystems is RECOMMENDED.

In MLS mode, be aware that *iso9660* filesystems do not support MLS labels on individual objects. You MAY use the `context=` mount option to specify an SELinux context including MLS level for the entire filesystem. It is RECOMMENDED that you use *iso9660* filesystems only for world readable data that does not need read protection.

Be aware that data written to removable media is not reliably protected by the DAC permission mechanism, and should be considered accessible to anyone with physical access to the media. It is RECOMMENDED to add the *ro* option to mount the file system read-only.

Note that these settings do not completely protect against malicious code and data, you MUST also verify that the data originates from a trustworthy source and does not compromise the server's security. Specifically, be aware of the following issues:

- Even unprivileged programs and scripts can contain malicious code that uses the calling user's rights in unintended ways, such as corrupting the user's data, introducing Trojan horses in the system, attacking other machines on the network, revealing confidential documents, or sending unsolicited commercial e-mail ("Spam").
- Data on the additional filesystem MUST have appropriate access rights to prevent disclosure to or modification by unauthorized users. Be aware that imported data could have been created using user names and permissions that do not match your system's security policies.
- You MUST NOT write data on removable file systems such as floppy disks, since it cannot be adequately protected by the system's access control mechanisms after being removed from the system. Please refer to section §4.2 "Backup and restore" of this guide for more information regarding non-filesystem-based backup.

Each new file system MUST be mounted on an empty directory that is not used for any other purpose. It is RECOMMENDED using subdirectories of */mnt* for temporary disk and removable storage media mounts.

For example:

```
# mount /dev/cdrom /mnt/cdrom -t iso9660 -o ro,nodev,nosuid,noexec
```

You MAY also add an equivalent configuration to */etc/fstab*, for example:

```
/dev/cdrom /mnt/cdrom iso9660 ro,noauto,nodev,nosuid,noexec 0 0
```

You MUST NOT include the *user* flag, ordinary users are not permitted to mount filesystems. This is also enforced by the deletion of the SUID bit on the *mount* command.

4.7 Configuration of kernel cryptographic support

The evaluated configuration blacklists the Intel AES-NI support kernel module. This blacklisting ensures that the AES-NI support is not loaded automatically.

You MUST NOT load this kernel module manually.

4.8 Encryption of partitions

RHEL provides the dm-crypt mechanism for setting up partitions where all data stored on those partitions are encrypted on the fly. When data is read from those partitions, the data is decrypted without any intervention by any user.

As the block device of the partition is subject to the cryptographic operation, there is no restriction which filesystem is used together with the encrypted block device. If you selected a full disk encryption or configured encryption for different partitions during the initial installation time as outlined in section §2.2.9 "Hard disk encryption", you already store data on dm-crypt protected hard disks.

You MAY configure yet unused or newly added hard disks or partitions using dm-crypt before creating a filesystem on them. The setup of a dm-crypt protected partition is performed using the `cryptsetup` application. Please refer to the *cryptsetup*(8) man page for instructions on using dm-crypt.

When using cryptsetup manually, you MUST use the LUKS extension and therefore the LUKS commands specified in *cryptsetup*(8).

The setup of a dm-crypt protected partition is performed with the *luksFormat* command to the `cryptsetup` application. You MAY use the *luksFormat* command with the option *–with-random* if your system is not in FIPS 140-2 mode of operation to ensure that the session key used to encrypt the data is derived from */dev/random* to ensure a very high quality of the key material:

```
        cryptsetup luksFormat --with-random /dev/sdd1
```

This option ensures that `cryptsetup` generates the key used for the cryptographic operations out of */dev/random* to ensure cryptographically strong keys. In FIPS 140-2 mode, this option is not necessary as `cryptsetup` uses the *libgcrypt* shared library for any cryptographic operation. The key material is generated out of an ANSI X9.31 DRNG which is seeded by using the device file linked to by */etc/gcrypt/rngseed*.

When you do not want to use the default cipher with *luksFormat* (see `cryptsetup -help` for the default), you MUST ensure that the following requirements are met when specifying the cipher:

Allowed ciphers:

- AES (128 bits, 192 bits, 256 bits)
- Serpent (128 bits, 192 bits, 256 bits)
- Twofish (128 bits, 192 bits, 256 bits)

Allowed block chaining modes:

- CBC
- GCM
- XTS

Allowed IV-handling mechanisms:

- GCM, XTS: plain
- GCM, XTS: plain64
- CBC: ESSIV
- GCM, XTS: benbi

After formatting, the *luksOpen* command has to be used to set up the encryption mechanism, i.e. to inform the kernel that any read and write operation is encrypted and decrypted on the fly. The device file created with the *luksOpen* command can now be used to create a file system which then can also be mounted.

For a regular operation, the *luksOpen* command has be used followed by a `mount` command with the device file created by *luksOpen*.

Please note that when the system was booted in FIPS 140-2 mode, the kernel does not allow the use of ciphers as well as hash mechanisms that are not FIPS 140-2 approved. Therefore, errors will occur when specifying non-approved ciphers or hashes.

4.9 Managing user accounts

4.9.1 Creating users

Use the *useradd*(8) command to create new user accounts, then use the *passwd*(1) command to assign an initial password for the user. Alternatively, if the user is present when the account is created, permit them to choose their own password. Refer to the manual pages for *useradd*(8) and *passwd*(1) for more information.

If you assign an initial password for a new user, you MUST transfer this initial password in a secure way to the user, ensuring that no third party gets the information. For example, you can tell the password to a user personally known to you. If this is not possible, you MAY send the password in written form in a sealed letter. This applies also when you set a new password for a user in case the user has forgotten the password or it has expired. You need to advise the user that he MUST change this initial password when he first logs into the system and select his own password in accordance with the rules defined in section §7.3 "Password policy" of this guide.

You MUST NOT use the `-p` option to *useradd*(8), specifying a password in that way would bypass the password quality checking mechanism. In addition, specifying a password on the command line makes this password temporarily visible to all users.

The temporary password set by the administrator MUST be changed by the user as soon as possible. Use the *chage*(8) command with the `-d` option to set the last password change date to a value where the user will be reminded to change the password. The RECOMMENDED value is based on the settings in */etc/login.defs* and is equivalent to today's date plus PASS_WARN_AGE minus PASS_MAX_DAYS.

Example:

```
useradd -m -c "John Doe" jdoe
passwd jdoe
chage -d $(date +%F -d "53 days ago") jdoe
```

The `-m` option to *useradd*(8) creates a home directory for the user based on a copy of the contents of the */etc/skel/* directory. Note that you MAY modify some default configuration settings for users, such as the default *umask*(2) setting or time zone, by editing the corresponding global configuration files:

```
/etc/profile
/etc/bashrc
/etc/csh.cshrc
```

The optional setting of labels and roles is discussed in §4.9.8 Defining user roles and MLS levels (MLS mode only).

4.9.2 Changing user passwords

If necessary, you MAY reset the user's password to a known value using `passwd` *`USER`*, and entering the new password. You cannot recover the previously used password, since the hash function used is not reversible.

4.9.3 SSH key-based authentication

The TOE allows the configuration of key-based authentication for SSH. Key-based authentication is configured on a per-user basis by managing the file *.ssh/authorized_keys* in the home directory of a user. For information on how to use that file, see sshd(8).

To generate DSA or RSA keys that can be used for key-based authentication, the tool *ssh-keygen*(8) is provided. As the SSH daemon only accepts SSH protocol version 2, only the protocol 2 keys are supported with the SSH daemon. Therefore, you MUST only use the option `-t rsa` or `-t dsa` when generating a key with *ssh-keygen*.

Please note that account locking does not prevent users to log onto the system with SSH key-based authentication.

4.9.4 Changing user properties

You MAY use the *usermod*(8) command to change a user's properties.

4.9.5 Locking and unlocking user accounts

Users MAY be locked out (disabled) using `passwd -l` *`USER`*, and re-enabled using `passwd -u` *`USER`*. Note that this locking only prevents password-based authentication attempts. SSH key-based authentication is unaffected by using `passwd -l`. To prevent SSH key-based logins, the file *.ssh/authorized_keys* located in the home directory of the user MUST be removed.

The *pam_faillock.so* PAM module enforces automatic lockout after excessive failed authentication attempts. Use the program *faillock* to view and reset the counter if necessary, as documented in the man page *faillock*(8). Note that the *pam_faillock* mechanism does not *prevent* password guessing attacks, it only prevents *use* of the account after such an attack has been detected. Therefore, you MUST assign a new password for the user before reactivating an account. For example:

```
# view the current counter value
faillock --user jdoe

# set new password, and reset the counter
passwd jdoe
faillock --user jdoe --reset
```

The *chage*(1) utility MAY be used to view and modify the expiry settings for user accounts. Unprivileged users are able to view but not modify their own expiry settings.

4.9.6 Removing users

The *userdel*(8) utility removes the user account from the system, but does not remove files outside the home directory (and the mail spool file), or kill processes belonging to this user. Use `kill` (or reboot the system) and `find` to do so manually if necessary, for example:

```
# Which user to delete?
U=jdoe

# Lock user account, but don't remove it yet
passwd -l $U

# Kill all user processes, repeat if needed (or reboot)
killall -u $U

# Recursively remove all files and directories belonging to user
# (Careful - this may delete files belonging to others if they
# are stored in a directory owned by this user.)
# Use the applicable file system type for your system.
find / -depth \( ! -fstype ext3 -prune -false \) \
        -o -user $U -exec rm -rf {} \;

# Remove cron jobs
crontab -u $U -r

# Now delete the account
userdel $U
```

Please note that similar concerns apply when a group is removed. The administrator MUST ensure that the files associated with the group are reassigned to other groups or deleted. In addition, the administrator MUST handle the processes currently executing with the deleted group.

In addition, the administrator should consider that the user ID or group ID may be used in ACLs where these ACLs should be checked for their validity.

If you need to create additional groups or modify or delete existing groups, use the *groupadd*(8), *groupmod*(8) and *groupdel*(8) commands.

You MAY assign group passwords and allow use of the *newgrp*(8) program to change groups. Note that the *gpasswd*(1) program will only work when run at *SystemLow* level.

4.9.7 Defining administrative accounts

Administrative users MUST be member of the *wheel* group. Specify the `-G wheel` option for the *useradd*(8) command when creating administrative users.

You MAY also use the *usermod*(8) command to change group membership. For example, if you want to add the user 'jdoe' to the *wheel* group, you could use the following:

```
# List the groups the user is currently a member of:
groups jdoe

# Add the additional group
usermod -G $(groups jdoe | sed 's/.*: //; s/ /,/g'),wheel jdoe
```

In MLS mode, administrative users MUST also be assigned to the *staff_u* user class, and MUST be assigned to one or more administrative roles. The *staff_u* user class gives permission to use the *sysadm_r*, *secadm_r*, and *auditadm_r* roles in addition to the default *staff_r* role. Use the following steps to define an administrative user in MLS mode:

```
useradd -m -c "John Doe" -G wheel jdoe
passwd jdoe
chage -m 1 -M 60 -W 7 jdoe
semanage login -a -s staff_u -r SystemLow-SystemHigh jdoe
restorecon -r /home/jdoe
```

4.9.8 Defining user roles and MLS levels (MLS mode only)

In MLS mode, use the *semanage*(8) program to assign SELinux user classes, roles, and MLS levels to users.

Here is an example of creating a nonadministrative user class with permission to access a range of MLS levels, then creating two users and assigning different clearances within that range to these users:

```
semanage user -a -R user_r -r s0-s4:c100.c299 -P user op_u

useradd -m op1
useradd -m op2
passwd op1
passwd op2
chage -m 1 -M 60 -W 7 op1
chage -m 1 -M 60 -W 7 op2
```

```
semanage login -a -s op_u -r s0-s2:c150.c159 op1
semanage login -a -s op_u -r s0-s3:c130.c249 op2
```

Please refer to the *semanage*(8) man page for more information about this program.

Section §4.17.3 "Creating a custom role (MLS mode only)" of this guide contains a more detailed example which includes defining a custom role and associated rights.

4.10 Using serial terminals

You MAY attach serial terminals to the system for use by system administrators.

Serial terminals are activated by adding an entry in the file */etc/inittab* for each serial terminal that causes *init*(8) to launch an *agetty*(8) process to monitor the serial line. *agetty* runs *login*(1) to handle user authentication and set up the user's session.

If you use serial terminals and require the MLS-compliant fail-safe audit mode, you MUST ensure that the file */etc/pam.d/login* is configured to use the *require_auditd* option for the *pam_loginuid.so* module in the `session` stack.

For example, adding the following line to */etc/inittab* activates a VT102-compatible serial terminal on serial port /dev/ttyS1, communicating at 19200 bits/s:

```
S1:3:respawn:/sbin/agetty 19200 ttyS1 vt102
```

The first field MUST be an unique identifier for the entry (typically the last characters of the device name). Please refer to the *agetty*(8) and *inittab*(5) man pages for further information about the format of entries.

You MUST reinitialize the *init* daemon after any changes to */etc/inittab* by running the following command:

```
init q
```

4.11 Managing data objects

4.11.1 Revoking access

As with most operating systems, access rights are checked only once, when the object is first accessed by the process. If the initial permission check was successful, read and/or write operations are permitted indefinitely without further checking, even if the access rights to the object are changed or revoked.

If this delayed revocation is not acceptable to you and you need to definitely ensure that no user processes are accessing an object after you have changed the access rights to that object, you MUST reboot the system. This ensures that no processes have open descriptors which could permit continued access.

4.11.2 SYSV shared memory and IPC objects

The system supports SYSV-compatible shared memory, IPC objects, and message queues. If programs fail to release resources they have used (for example, due to a crash), the administrator MAY use the *ipcs*(8) utility to list information about them, and *ipcrm*(8) to force deletion of unneeded objects. Note that these resources are also released when the system is rebooted.

For additional information, please refer to the *msgctl*(2), *msgget*(2), *msgrcv*(2), *msgsnd*(2), *semctl*(2), *semget*(2), *semop*(2), *shmat*(2), *shmctl*(2), *shmdt*(2), *shmget*(2) and *ftok*(3) manual pages.

4.11.3 Posix Message Queues

POSIX message queues are supported as an alternative to SYSV message queues. Users and administrators MAY use the system calls and corresponding library functions documented in the *mq_overview*(7) man page, such as *mq_open*(2) and *mq_unlink*(2).

The message queue filesystem (type *mqueue*) MAY be mounted in case filesystem-based access to POSIX message queues is requested.

4.12 Configuring object access rights

Administrators MAY use the *chown*(1), *chgrp*(1), and *chmod*(1) tools to configure DAC access rights in Base and MLS mode. You MUST NOT grant additional access to objects that are part of the evaluated configuration.

In MLS mode, administrators MAY use the *chcon*(1) tool to change the MLS level and SELinux type of objects. You MUST NOT grant additional access to objects that are part of the evaluated configuration. The *chcat*(8) tool is unsuitable for MLS mode and MUST NOT be used.

Please refer to the respective man pages for more information about these tools.

4.13 Setting the system time and date

You MUST verify periodically that the system clock is sufficiently accurate, otherwise log and audit files will contain misleading information. When starting the system, the time and date are copied from the computer's hardware clock to the kernel's software clock, and written back to the hardware clock on system shutdown.

All internal dates and times used by the kernel, such as file modification stamps, use universal time (UTC), and do not depend on the current time zone settings. Userspace utilities usually adjust these values to the currently active time zone for display. Note that text log files will contain ASCII time and date representations in local time, often without explicitly specifying the time zone.

The *date*(1) command displays the current time and date, and can be used by administrators to set the software clock, using the argument *mmddHHMMyyyy* to specify the numeric month, day, hour, minute and year respectively. For example, the following command sets the clock to May 1st 2004, 1pm in the local time zone:

```
date 050113002004
```

The *hwclock*(8) can query and modify the hardware clock on supported platforms, but may not available in virtual environments. The typical use is to copy the current value of the software clock to the hardware clock. Note that the hardware clock MAY be running in either local time or universal time, as indicated by the *UTC* setting in the */etc/sysconfig/clock* file. The following command sets the hardware clock to the current time using UTC:

```
hwclock -u -w
```

Use the command *tzselect*(8) to change the default time zone for the entire system. Note that users MAY individually configure a different time zone by setting the *TZ* environment variable appropriately in their shell profile, such as the *$HOME/.bashrc* file.

4.14 Firewall configuration

You MAY enable, reconfigure, or disable the builtin network firewall as required. RHEL allows the following types of firewall configuration to control traffic:

- The packet filtering of IP, TCP, UDP, ICMP protocols is implemented with the `iptables` command which uses kernel support for controlling traffic. Iptables can be used to set up very small and lean packet filter rules. On the other hand, very complex and sophisiticated filtering rules can be configured as well to suit the need of the administrator. An elaborate introduction is given in the Red Hat Enterprise Linux 6 Security Guide section 2.6 accessible at http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/6/html/Security_Guide/index.html. In addition, *iptables*(8) discusses use of the application.
- RHEL allows the configuration of virtual machines using the KVM functionality and connects the guest software with external networks using the Linux kernel's bridging functionality. As the bridging functionality is enforced at Ethernet layer, the Linux kernel does not engage its TCP/IP stack for packets travelling through the bridge to received by either some KVM guest software or remote systems. RHEL provides a packet filter mechanism for filtering packets at the Ethernet layer using the `ebtables` functionality. The man page *ebtables*(8) discusses the concept as well as the use of the packet filter.

4.14.1 iptables auditing of packet filter operations

Iptables is extended by an *AUDIT* target that conceptually works similar to the *LOG* target documented in *iptables*(8).

This target allows the logging of matching packets with the audit facility. When this option is set for a rule, the Linux kernel will generate an audit entry for each matching packet which holds details about the packet depending on the protocol of the packet.

Data that is always logged includes the action applied to the packet (*action=* audit information), the inbound interface (*inif=*), and the outbound interface (*outif=*).

If the matched packet from the ARP layer, the source (*smac=*) and destination (*dmac=*) MAC address as well as the protocol type (*macproto=*) is logged.

When using the target with IPv4 and IPv6 packets, the source (*saddr=*) and destination (*daddr=*) IP address, as well as the protocol type (*proto=*) is logged. In addition, the audit trail identifies whether the packet is truncated (*truncate=1*) or whether it is fragmented (*frag=1*).

In case the IP packet encapsulates TCP or UDP data, the source (*sport=*) and destination (*dport=*) port information is audited.

If the IP packet encapsulates ICMP data, the ICMP type (*icmptype=*) and the ICMP code (*icmpcode=*) is logged.

The following option is allowed with the *AUDIT* target:

- `-type` *type* sets the action type to be audited. Valid options are either *accept*, *drop*, or *reject*.

4.14.2 ebtables auditing of packet filter operations

Similarly to iptables, ebtables is extended by an *AUDIT* watcher. The implementation of the *AUDIT* watcher for ebtables is identical to the *AUDIT* target of the iptables mechanism.

This watcher allows the logging of matching frames with the audit facility. When this option is set for a rule, the Linux kernel will generate an audit entry for each matching frame which holds details about the frame depending on the contents of the frame.

The audit trail generated by the *AUDIT* watcher is identical to the information specified in §4.14.1 "iptables auditing of packet filter operations".

Also, the options allowed with the *AUDIT* watcher are identical to the listed options specified in §4.14.1 "iptables auditing of packet filter operations".

4.14.3 KVM ARP spoofing and poisoning prevention

The `libvirtd` virtual machine management daemon allows the configuration of of an ARP spoofing and poisoning prevention schema. When enabling this configuration for a virtual machine, only outgoing communication with a source ARP and IP addresses that is configured for the virtual machine is allowed. Conversely, for inbound communication to a virtual machine, the communication must have the destination ARP and IP addresses of the virtual machine as set by `libvirtd`.

`libvirtd` uses the `ebtables` and `iptables` commands to set up the protection mechanism.

The support is enabled with the following XML setting that can either be set in the XML configuration file for a virtual machine or sent to `libvirtd`:

```
<interface type='network'>
   ...
   <mac address='aa:bb:cc:dd:ee:ff'/>
   <ip address='1.2.3.4'/>
   <filterref filter='clean-traffic'/>
   ...
</interface>
```

You MUST specify the MAC address and IP address to be assigned to the virtual machine. `libvirtd` will invoke QEMU with the MAC address which in turn is handed off to the guest operating system. The IP address is set by either the `dnsmasq` DNS server or must be manually configured with the guest operating system.

To ensure that the virtual machine communication is fully protected against ARP spoofing and poisoning attacks, the administrator MUST consider the following requirement. Even when only one virtual machine is deemed critical and ARP spoofing and poisoning preventioning support shall be configured for this particular virtual machine only, the following MUST be configured: All virtual machines that execute on the host must have the above configuration set. This ensures that the ARP tables of other guest systems cannot be altered by another virtual machine.

Background information on the ARP spoofing and poisoning protection schema can be found at http://berrange.com/posts/2011/10/03/guest-mac-spoofing-denial-of-service-and-preventing-it-with-libvirt-and-kvm/.

4.15 Screen saver configuration

The `screen` application is used to provide a locking mechanism of the current terminal for every user. The `screen` locking is invoked by the following means:

- The locking is executed automatically after a period of inactivity on the terminal defined by a timeout in either */etc/screenrc* or *~/.screenrc* using the *lockscreen* configuration value.
- Every user can lock his screen by executing the `C-a C-x` screen key binding combination.

You MAY change the timeout value for locking the session in */etc/screenrc* with the value for *lockscreen*.

Please note that users can modify the timeout by providing their own *~/.screenrc*. You can disable the support for per-user configuration files by invoking screen with the option of *-c /dev/null*.

To invoke `screen` automatically upon log in, you MAY enter the following line to either */etc/bash_profile* for system-wide enforcement or *~/.bash_profile* for a per-user enforcement. Note that a user can change *~/.bash_profile*.

```
exec screen
```

4.16 OpenSSH configuration

The evaluated configuration requires that keys generated for OpenSSH applications including the `sshd` daemon, the `ssh` client application and `ssh-keygen` must be generated using a random number generator that is seeded with at least 48 bits of entropy.

OpenSSH uses the OpenSSL deterministic random number generator for generating keys. This deterministic random number generator is seeded by reading the seed from */dev/random*. The seeding process is described in the configuration file */etc/sysconfig/sshd*.

Using */dev/random* in the evaluated configuration deviates from the default behavior of the OpenSSH applications which per default employs */dev/urandom*. The difference is that */dev/random* blocks until sufficient entropy is available to satisfy the request for entropy. This implies that the OpenSSH applications block when they seed or re-seed until the requested amount of entropy is delivered.

The blocking may cause irritation by a user as the application seemingly stalls without any notification. The administrator is advised to inform the user base about this blocking behavior.

The blocking behavior is controlled with the environment variable *SSH_USE_STRONG_RNG* which must be set according to the following rules:

- For the `sshd`, the environment variable must be set in */etc/sysconfig/sshd*. After setting, the `sshd` daemon must be restarted using the start script.
- For applications invoked by users, the file */etc/profile/cc-configuration.*sh* contains the setting of this environment variable for Bourne shell and derivatives as well as C-shells and its derivatives.

In addition to the use of the proper random number generator, the AES-NI support provided with OpenSSL must be deconfigured for the OpenSSH applications. The AES-NI support was not subject to this current evaluation and can therefore not be used to encrypt the communication channel. The AES-NI support is disabled for OpenSSL by setting the *OPENSSL_DISABLE_AES_NI* environment variable. The evaluated configuration already sets this environment variable in */etc/sysconfig/sshd* for the `sshd` daemon. Also, the environment variable is set in */etc/profile.d/cc-configuration.*sh*.

4.17 SELinux configuration

4.17.1 General SELinux configuration

In MLS mode, SELinux MUST be enabled and in enforcing mode, and MUST use the "mls" policy. The */etc/selinux/config* file MUST have the following content:

```
SELINUX=enforcing
SELINUXTYPE=mls
```

In Base mode, the evaluated configuration keeps the SELinux system enabled in a static configuration, but does not depend on SELinux for any security features. You MAY modify the SELinux configuration, for example to add additional restrictions.

In Base mode, the */etc/selinux/config* file has the following content by default:

```
SELINUX=enforcing
SELINUXTYPE=targeted
```

In Base mode, you MAY disable SELinux by using one of the settings `SELINUX=disabled` or `SELINUX=permissive` instead, or configure a different policy, but any additional restrictions added by SELinux are beyond the scope of the Base configuration. (Note that reconfiguring the SELinux policy is likely to affect your support contract status. This is also beyond the scope of this document.)

4.17.2 MLS mode specific policy module

The *lspp_policy* SELinux policy module adds permissions necessary for correct system operation in MLS mode. You MAY customize some of the settings in the */usr/share/selinux/devel/lspp_policy.te* file as indicated in the file comments:

```
## Customized SELinux policy for MLS evaluated configuration

policy_module(cc_mls_policy,1.0)

################################################################################
### Additional audit
################################################################################

gen_require(`
        attribute domain;
')

# Audit setting of security relevant process attributes
# These settings are OPTIONAL
auditallow domain self:process setcurrent;
auditallow domain self:process setexec;
auditallow domain self:process setfscreate;
auditallow domain self:process setsockcreate;
```

After any changes to this file, use the following steps to reload the module:

```
cd /usr/share/selinux/devel/

# as role "sysadm_r":
make cc_mls_policy.pp

semodule -i cc_mls_policy.pp
```

4.17.3 Creating a custom role (MLS mode only)

This example shows how to create a "backup admin" role with the privilege to read all files on the system, but no special write privileges. This role could be used to perform system backups, but without the risk of overwriting or modifying any system files.

The role created in this example is an administrative role. The people assigned to this role are required to follow the same operational rules as all administrators, and are also assumed to be fully trusted not to undermine system security.

The root password is needed to use the role, but the actions available to the backup admin are limited while using the role. For example, changing roles to "sysadm_r" is denied when using an interactive ssh session. However, if the backup admin has direct console access, the root password would permit a login as "root" directly with unlimited administrative actions.

As a first step, define a SELinux policy module defining the role and its privileges. Do the following steps as role "sysadm_r":

```
## role "sysadm_r"

# change to the local policy directory
cd /usr/share/selinux/devel

# create the new policy module source
cat <<-'_EOF'_ > backupadm.te
        policy_module(backupadm,1.0)

        gen_require(`
                role staff_r;
                type staff_t, staff_devpts_t, staff_tty_device_t;
        ')

        # Define the role and domain
        userdom_unpriv_user_template(backupadm)
        role backupadm_r types backupadm_t;

        # Allow members of staff_r to transition to this role
        userdom_role_change_template(staff, backupadm)

        # grant DAC read override capability
        allow backupadm_t self:capability dac_read_search;

        # grant MLS read override capability
        mls_file_read_up(backupadm_t)

        # grant RBAC file read override capability
        files_read_all_files(backupadm_t)
_EOF_

# build binary policy module
make backupadm.pp

# Assign the default domain for the role
echo "backupadm_r:backupadm_t" >>/etc/selinux/mls/contexts/default_type
```

If necessary, as "sysadm_r", create an administrative user account for the user who will be using this role:

```
## role "sysadm_r"
useradd -G wheel jdoe
passwd jdoe
```

Insert the new policy module into the active policy:

```
## role "sysadm_r"
semodule -i backupadm.pp
```

Now create a new SELinux user class with the right to use the new role (but not any other administrative roles), and assign the user(s) to this user class to give them the right to use the role:

```
## role "sysadm_r"
semanage user -a -R "staff_r backupadm_r" -P backupadm backup_u
semanage login -a -s backup_u -r SystemLow jdoe
```

To test the new role, log in as this user, "su" to root, and enter the new role:

```
ssh jdoe@localhost
        /bin/su -
        newrole -r backupadm_r
```

Verify that you can read all files, but not modify any system files:

```
## user "jdoe", role "backupadm_r"
cat /etc/shadow     # succeeds
touch /etc/shadow   # fails
```

To delete the new role, first ensure that no users are mapped to this role (the system will refuse to remove the module from the policy if the role is in use), and remove the module:

```
## role "sysadm_r"
semanage login -d jdoe
semanage user -d backup_u
semodule -r backupadm
```

4.17.4 Defining hierarchical roles (MLS mode only)

Hierarchical roles MAY be defined using the *dominates* operator.

This example defines a *root_r* role that combines the rights of *sysadm_r*, *secadm_r*, and *auditadm_r*:

```
# as role "sysadm_r"

cd /usr/share/selinux/devel

### set up the new policy module
cat <<'_EOF_' > root_role.te
        policy_module(root_role,1.0)

        gen_require(`
                role sysadm_r, secadm_r, auditadm_r;
        ')
```

```
                # Define the role and domain
                userdom_admin_user_template(rootuser)

                # Allow members of staff_r to transition to this role
                userdom_role_change_template(staff, rootuser)

                # define new role in terms of existing roles
                dominance { role rootuser_r {
                     role sysadm_r;
                     role secadm_r;
                     role auditadm_r;
                } }
        _EOF_

        # build binary policy module
        make root_role.pp
```

If necessary, as "sysadm_r", create an administrative user account for the user who will be using this role:

```
## role "sysadm_r"
useradd -G wheel jdoe
passwd jdoe
```

Insert the new policy module into the active policy:

```
## role "sysadm_r"
semodule -i root_role.pp
```

Now create a new SELinux user class with the right to use the new role (but not any other administrative roles), and assign the user(s) to this user class to give them the right to use the role:

Now assign the new role to a user:

```
## role "sysadm_r"
semanage user -a -R "staff_r rootuser_r" -P staff rootuser_u
semanage login -a -s rootuser_u jdoe
```

The new administrative user may then switch freely among the subsidiary roles without having to list these roles separately in the login mapping:

```
# as user in "root_u" class:
newrole -r auditadm_r
```

Changes to the *root_r* role definition, such as adding new subsidiary roles, will automatically change the rights of all users in this class.

4.18 Labeled networking (MLS mode only)

In MLS mode, you MUST use one of the two available labeled networking mechanisms to ensure that the data flow restrictions are properly enforced when using networking related system call interfaces.

4.18.1 IPSec labeled networking

Setting up IPSec is described in the *ipsec*(8), *ipsec.conf*(5), and *ipsec.secrets*(5) man pages.

The encryption and authentication properties of IPSec are beyond the scope of this guide and evaluation. It is concerned only with the use of IPSec to transport MLS labels.

Be aware that the labeled IPSec configuration may result in being unable to access the system using the network. It is RECOMMENDED to do this when logged in at the local system console.

You MUST either define a pre-shared key for each system that will communicate using labeled IPSec or use certificates as documented in *ipsec.secrets*(5).

To label a particular IPSec communication channel with one particular label, the keyword "policy_label=" is to be used in the `pluto` configuration file of */etc/ipsec.d/ipsec.conf*. The security label specified in *ipsec.conf* is associated with IPsec policies in the Security Policy Database (SPD). This Policy label is not exchanged between IKE peers. Security label associated with traffic stream is associated with IPsec SAs, and this label is exchanged in IKE messages. The traffic stream security label obtained from remote IPsec peer is verified against the local policy label using SElinux APIs. There can be multiple SA labels (traffic streams each with its own security label different from others) associated with one policy label. SA labels should comply with policy label, otherwise SA establishment will not succeed.

After the configuration, you MUST enable the configuration by triggering an SA establishment using the following command:

```
        ipsec auto --route <conn-name>
```

where *conn-name* is the label of the connection specified in *ipsec.conf*.

The following example describes how to set up labeled IPSec between two machines with IP addresses 172.16.2.55 and 172.16.2.66.

On both systems, create the file */etc/ipsec.secrets* containing either pre-shared secret keys, or certificates as discussed in *ipsec.secrets*(5).

On the machine with the IP address of 172.16.2.55, create the file */etc/ipsec.d/ipsec.conf* with the following contents:

```
    conn label
        auto=add
        authby=secret
        type=transport
        left=172.16.2.55
        right=172.16.2.66
        ike=3des-sha1
        phase2=ah
        phase2alg=sha1
        labeled_ipsec=yes
        policy_label=system_u:object_r:ipsec_spd_t:s0-s15:c0.c1023
```

Conversely, on the machine with the IP address of 172.16.2.66, create the file */etc/ipsec.d/ipsec.conf* with the following contents:

```
    conn label
        auto=add
        authby=secret
        type=transport
        left=172.16.2.66
        right=172.16.2.55
```

```
ike=3des-sha1
phase2=ah
phase2alg=sha1
labeled_ipsec=yes
policy_label=system_u:object_r:ipsec_spd_t:s0-s15:c0.c1023
```

Please set the contents of the "policy_label=" configuration value to the appropriate SELinux label that you want to associate with the connection.

Then trigger an SA establishment as outlined above. Note, the SA may be set up at start time of `pluto` when using the configuration of *auto=route* instead of *auto=add*. For more information, please see the referenced man pages.

To disable labeled networking and resume unlabeled networking operations, disable the `pluto` daemon.

4.18.2 CIPSO labeled networking

Use the following commands to activate CIPSO labeled networking:

```
netlabelctl cipsov4 add pass doi:1 tags:1
netlabelctl map del default
netlabelctl map add default protocol:cipsov4,1
netlabelctl unlbl accept off
```

Be aware that the `unlbl accept off` configuration may result in being unable to access the system using the network. It is RECOMMENDED to do this when logged in at the local system console.

You MAY define other DOI settings as defined in the *netlabelctl*(8) man page.

Use the following commands to disable CIPSO labeled networking:

```
netlabelctl unlbl accept on
netlabelctl map del default
netlabelctl map add default protocol:unlbl
```

4.18.3 xinetd configuration for labeled networking

When labeled networking is enabled (using either CIPSO or labeled IPSEC), you MUST disable *sshd* on the default port 22 by issuing the following command:

```
chkconfig sshd off
```

SSH sessions established by users will automatically run at the level corresponding to the label of the network data. By default, the label aware *sshd* instance runs on TCP port 222, you MAY change the port number by editing the */etc/xinetd.d/sshd-mls* file, for example by setting `port=22` to replace the disabled non-label-aware *sshd*. You MUST register the port number using the following command (using the appropriate port number instead of "222"):

```
semanage port -a -t ssh_port_t -p tcp 222
```

The *sshd* server will enforce that the requested level is within the permitted range for that user, but cannot control any information flow happening within the *ssh* client itself. You MUST activate labeled networking to help ensure proper labeling of information across the system boundary.

During the configuration of labeled networking you MUST disable the non-label-aware stand-alone *sshd* daemon. Failure to disable this daemon implies that users can change labels and therefore establish a communication path between labels, violating the MLS rules.

4.19 Remote audit configuration

The audit mechanism supports a configuration where audit data is stored with a remote server. The auditd daemon uses the services of the audispd daemon. That daemon in turn uses the services of audisp-remote which handles the offloading of data to a remote server.

In addition to the support for submitting audit data to remote systems, the auditd daemon can be configured to listen on the network to receive audit data from a remote audisp-remote application.

Please note that the yet disabled configuration for receiving data shipped with the evaluated configuration is synchronized with the disabled and shipped configuration for the submission of audit data. You MAY use the given configuration examples for a valid configuration of the sending and receiving endpoints.

4.19.1 Configuration of sending audit data

The configuration for sending audit data to a remote auditd daemon is performed by performing the following configurations:

- Configure the use of the audispd daemon by auditd: enable the configuration value *dispatcher* to point to audispd in */etc/audit/auditd.conf*.
- Enable the audisp-remote plugin by changing the configuration variable *active* to *yes* in */etc/audisp/plugins.d/au-remote.conf*.
- Ensure that the network settings in */etc/audisp/audisp-remote.conf* point to the remote network server with the receiving auditd.
- Ensure that the remaining configurations in */etc/audisp/audispd.conf* as well as */etc/audisp/plugins.d/au-remote.conf* are appropriate.

Please restart the auditd daemon.

The configuration of the remote auditing support is independent from keeping a local audit trail. Therefore, the local audit trail configuration can be disabled or enabled while the remote audit configuration is set up.

Please refer to the man pages of *auditd.conf*(5), *audisp-remote.conf*(5), and *audispd.conf*(5) for more details.

4.19.2 Configuration of receiving audit data

The auditd daemon implements the support for receiving audit data from the network. Several configuration options in */etc/audit/auditd.conf* must be set to define the networking behavior of auditd. Please see the yet disabled example configuration provided in */etc/audit/auditd.conf* supported by *auditd.conf*(5) for details.

Chapter 5

Kernel-based virtual machine (KVM) management

The TOE provides virtual machine environments which allow other operating systems to execute concurrently with the RHEL host system. Each virtual machine is represented as a process in the RHEL host system and is subject to the standard Linux constraints for processes. The virtualization and simulation logic that supports the operation of a virtual machine executes within the same process of the respective virtual machine. When operating virtual machines, the RHEL host kernel acts as the hypervisor for the guest systems.

Treating each virtual machine as a normal process by the RHEL host system allows the concurrent execution of standard applications at the same time. Administrative tools are implemented as standard Linux applications. In addition, the computing environment provided by the RHEL host system to users logging into that system does not differ from a standard RHEL system even while virtual machines execute. Therefore, standard Linux applications and daemons may operate concurrently with virtual machines. To access the RHEL host system, the TOE provides console access via OpenSSH as well as access via the virtual machine management daemon provided by `libvirtd` via OpenSSH. In addition, the virtual machine consoles can be accessed using VNC using an already established OpenSSH channel. The VNC server is implemented by the `QEMU` virtualization and simulation logic. The use of OpenSSH for accessing the `libvirtd` management daemon as well as VNC is encapsulated in different client applications which are not part of the TOE, like `virsh` and `virt-manager`.

When using RHEL as a host system for virtual machines, different UIDs and GIDs are used to separate different users, services, or virtual machines provided by the system. In such a case, it is assumed that these processes are responsible for the safeguarding of their data. Note: the evaluated configuration permits the use of the host system for regular operation by normal users at the same time when virtual machines execute. It depends on the administrator-defined policies whether such a dual use is allowed.

Hosting virtual machines with guest operating systems increase the availability requirements on the host system. Without guest systems, a failing RHEL instance renders the services offered by the RHEL system offline. However, a failing RHEL system with a number of guest systems terminates the services offered by RHEL and all hosted guest systems. The impact of such a discontinuity is increased by a several orders of magnitude. Therefore, you MUST plan the deployment of RHEL as a host system for other operating systems carefully to prevent service discontinuities affecting the overall goals of your IT infrastructure.

The following sections discuss the configuration of virtual machines and their resources.

5.1 Hardware configuration

The hardware covered by this evaluation provides all support needed for KVM. However, you MUST ensure that the virtualization hardware mechanisms are enabled. It MAY be possible that they are disabled using BIOS settings. Please check your hardware manuals or hardware vendor to ensure that the following hardware support is enabled:

Intel x86_64 bit based systems

- The processor's Virtualization Technology extensions (VT-x) support must be enabled. If that support is enabled, the following command returns with a listing of processor capabilities:

  ```
  cat /proc/cpuinfo | grep vmx
  ```

- The processor's Enhanced Page Tables (EPT) support must be enabled. If that support is enabled, the following command returns with a listing of processor capabilities:

  ```
  cat /proc/cpuinfo | grep ept
  ```

AMD Opteron based systems

- The AMD-V processor's support for virtual machines must be enabled. If that support is enabled, the following command returns with a listing of processor capabilities:

  ```
  cat /proc/cpuinfo | grep svm
  ```

- The processor's Nested Page Tables (NPT) support must be enabled. If that support is enabled, the following command returns with a listing of processor capabilities:

  ```
  cat /proc/cpuinfo | grep npt
  ```

5.2 Kernel-based Virtual Machine (KVM) configuration

The evaluated configuration allows the administration of virtual machines. The management daemon of `libvirtd` as well as a virtual machine's console are accessed via SSH.

The `libvirtd` virtual machine management daemon allows the specification of virtual machines, and the assignment of resources to these virtual machines. Once a virtual machine is configured, this daemon also allows the starting and stopping of a virtual machine.

Another aspect of `libvirtd` is the configuration of the host system and its resources which are made available to virtual machines, including the configuration of networks and storage areas.

The functionality mentioned for `libvirtd` allows users to define and start virtual machines. However, access to the virtual machine's console is established separately. The virtual machines are instantiated as a Linux process using the `QEMU` virtualization support. `QEMU` emulates various devices for a virtual machine, including the console devices with keyboard, mouse and display. These simulated devices are translated into a VNC server that the owner of the virtual machine can connect to.

Access to the `libvirtd` management daemon and the VNC server instantiated for a virtual machine by `QEMU` is established via SSH. Virtual machine client applications that are distributed with RHEL can be used to access both, the `libvirtd` management daemon and the VNC server. For example, the client applications of `virsh` or `virt-manager` allow the access of both components via SSH. Note that both client applications are not covered by the evaluation.

Users managing virtual machines and who are allowed to access the VNC servers MUST have a local account on the TOE and MUST be members of the *libvirt* group. These users do not need any other elevated privileges like root access. Before these users are able to access `libvirtd` and the VNC servers they must authenticate via SSH using their accounts. To access the `libvirtd` virtual machine management daemon remotely via SSH, use the following command:

```
virsh connect qemu+ssh://USER@HOST/system
```

Replace the strings USER with the user ID you want to authenticate with and HOST with the hostname or IP address of the system executing the `libvirtd` management daemon.

To access `libvirtd` locally on the TOE system, you MAY use the following command:

```
virsh connect qemu:///system
```

In this case, the `virsh` command directly accesses the Unix domain sockets provided by `libvirtd` in */var/run/libvirt/*.

To manage virtual machines and resource assignments, please see the *virsh*(1) man page. This command allows the administration of virtual machines, their resources, and maintaining the life cycle of virtual machines.

5.2.1 Virtual machines started outside libvirtd

In the TOE version, only virtual machines instantiated with the `libvirtd` virtual machine management daemon are covered with the separation mechanisms that were subject to evaluation. Normal users MAY start virtual machines outside of `libvirtd`. These virtual machines have the same privileges on the host system as the user that started them.

Users may invoke the `QEMU` application with similar command line options as the `libvirtd` daemon. In this case, the virtualization environment starts and the guest operating system may become active. However, this startup does not provide any assurance evaluated with the CC evaluation. In particular, the following support for virtual machines is missing rendering the execution of such virtual machines less effective:

- The hardware virtualization support MAY not used as the Linux kernel interface allowing user space to utilize the hardware support MAY not be accessible to normal users. The device file */dev/kvm* provides access to the hardware support.
- The separation mechanism between multiple instances of virtual machines as discussed in §5.3.2 "SELinux dynamic labeling" is not set up and enforced.

To ensure that all support mechanisms for the proper operation and separation of virtual machines are employed, virtual machines MUST be configured and managed using the `libvirtd` virtual machine management daemon.

5.2.2 System versus session instances of libvirtd

The `libvirtd` virtual machine management daemon has the capability to provide unprivileged users (i.e. users not part of the *libvirt* group) to manage virtual machines. In this scenario, a `libvirtd` instance executes with the user ID of the user. This operational mode is called *session* mode.

Contrary, the privileged instance of `libvirtd` is known as *system* instance.

The operational mode that the connecting user wants to access is supplied with the URI when connecting to `libvirtd`. The URI listed in §5.2 "Kernel-based Virtual Machine (KVM) configuration" contains the string *system* and therefore connects to the privileged instance of `libvirtd`.

The virtual machines started from the *session* instance of the `libvirtd` virtual machine management daemon lack the same capabilities as virtual machines that are started completely outside the `libvirtd` daemon. Therefore, the capabilities listed in §5.2.1 "Virtual machines started outside libvirtd" are missing for virtual machines started from the *session* instance of `libvirtd` as well.

This guidance document and all described virtual machine capabilities apply only to virtual machines controlled by the *system* instance of `libvirtd`.

5.3 Separation of virtual machines

The `libvirtd` virtual machine management daemon provided by RHEL ensures that the execution environment as well as the resources allocated to a specific virtual machine are separated from other virtual machine instances. Such a transparent setup aids the administrator in keeping a secure configuration for the host system as well as the virtual machines.

5.3.1 Unprivileged user and group IDs

The host system protects its operation against the virtual machines by executing them with a fully unprivileged user and group ID, the ID *qemu*. This user ID has equivalent rights to normal unprivileged users and can therefore not harm the host system even if the guest operating system would be able to exploit any vulnerabilities in the hardware emulation code provided with the `QEMU` application. The `libvirtd` virtual machine management daemon ensures that the processes implementing the virtual machine environment are started with the *qemu* user and group ID. The specification of these IDs is given in */etc/libvirt/qemu.conf*.

The configuration option option that enables the startup of a virtual machine with the configured user ID and the modification of the ownership of the file system resources is found in */etc/libvirt/qemu.conf* by setting the configuration value *dynamic_ownership* to 1.

5.3.2 SELinux dynamic labeling

In addition to the protection of the host system from the virtual machines, the host system uses another mechanism to ensure full separation between virtual machines. As all virtual machines execute with the same user ID and group ID it is possible in theory that these virtual machine processes may interfere with each other. The SELinux targeted policy distributed with RHEL implements a strict separation between virtual machines and its resources.

With the sVirt mechanism implemented with the `libvirtd` virtual machine management daemon, the SELinux category is automatically generated. These sVirt-generated unique SELinux categories are assigned to each virtual machine and its resources. The SELinux policy prevents any access request by a virtual machine to a resource if the SELinux category does not match. In the evaluated configuration, the unique SELinux category for each virtual machine is calculated automatically by the `libvirtd` virtual machine management daemon during startup of the virtual machine.

The virtual machine process as well as all resources assigned to the virtual machine via the `libvirtd` management framework are re-labeled accordingly by `libvirtd`.

More information about the dynamic labeling can be found at *http://libvirt.org/drvqemu.html#securitysvirt*.

5.3.3 SELinux static labeling

In the evaluated configuration, dynamic labeling is per default activated to achieve full separation of the virtual machine. However you MAY use static labeling as documented below in Base mode. Contrary, in MLS mode you MUST use static labeling as `libvirtd` does not modify the label on the fly.

If the administrator wants to set the label of the virtual machine process and the resources manually, the *dynamic_ownership* must be set to 0. In this case, the SELinux label is specified in the XML descriptor of the virtual machine in */etc/libvirt/qemu/*. The label specification may look as follows:

```
<seclabel type=’static’ model=’selinux’>
    <label>system_u:system_r:svirt_t:s0:c77,c525</label>
</seclabel>
```

The *label* attribute sets the label of the virtual machine process. If static labeling is used, you MAY only modify the category part (i.e. the "c77,c525" setting of the example). You MUST NOT modify any other part of the SELinux label. When changing other parts of the SELinux label, the full separation of virtual machines cannot be ensured any more.

In addition, when setting static labels, you MUST ensure that the category given to a virtual machine is unique and not used by other virtual machines.

Also, you MUST use the *selinux* security model as given in the above mentioned example.

Prior to starting the virtual machine, you MUST manually set the label on the resources configured for the virtual machine. These labels must ensure that the virtual machine process can access the resource. The labeling of these resources must be performed as outlined in *http://libvirt.org/drvqemu.html#securitysvirt*.

The static labeling allows the administrator the most flexibility on setting the permissions on the virtual machine. However, it also places the burden of ensuring proper virtual machine separation on the administrator.

5.4 Networking considerations

Virtual machines MAY be configured with one or more network interfaces. The `libvirtd` virtual machine management daemon allows the configuration of the networking capabilities for virtual machines.

Two aspects MUST be configured to establish the network connectivity for virtual machines:

1. The `libvirtd` MUST be configured with one or more networks. Each network configuration results in an automated setup of a bridge network interface on the host system. That bridge interface is assigned with a physical network interface on the host. To achieve separation between these networks, each network MUST be assigned to an individual physical network interface.
2. Each virtual machine network configuration must be assigned to belong to one network defined for `libvirtd`. With this link, the virtual machine becomes part of the bridge defined for the network interface.

A network bridge provided with the host system can be considered to resemble a network switch. Each member of the bridge is able to directly communicate with each other member. Therefore, if virtual machines shall be configured that must not communicate with each other, separate networks must be specified with `libvirtd`. These networks must be assigned with different physical network interfaces on the host. The network interfaces of the virtual machines that shall be isolated must be assigned different networks.

When `libvirtd` instantiates a virtual machine, it sets up a TAP device connected to the bridge the virtual machine is configured to have access to. The file descriptor of that TAP device is transferred to the `QEMU` process by the `libvirtd` virtual machine management daemon as the unprivileged `QEMU` is not allowed to open the TAP device. With this file descriptor, the virtual machine process is now able to implement all network protocols starting with OSI-layer 2.

As `libvirtd` provides virtual machine processes with a TAP device to the host system's bridge interface, a virtual machine is able to generate full Ethernet communication including sniffing the network or flooding the network with data. This implies that although the host system fully isolates the virtual machines from the host system as well as from each other, virtual machines have the same network capabilities to the assigned network as bare-metal systems connected to the same network! Therefore, guest software executing within the virtual machines must be as trustworthy as individual physical machines on the same network. KVM does not and is not designed to stop rogue guest software from misusing access to the networks they are allowed to access.

5.4.1 Packet filtering

The host system provides packet filtering functionality for the bridges implemented with *ebtables*(8). Please see the man pages for details about setting up packet filters on bridges.

More information about ebtables is presented in section §4.14 "Firewall configuration".

5.5 Device assignment

Together with the support of the *libvirtd* management daemon, KVM provides the functionality of assigning PCI as well as USB devices to virtual machines for their exclusive use. A device can only be assigned to one virtual machine using this mechanism.

The following sections discuss the device assignment mechanisms separately for PCI and USB devices.

5.5.1 PCI device assignment

PCI device assignment MUST NOT be configured as it is considered to be unsafe in secure environments.

5.5.2 USB device assignment

Unlike the PCI device assignment, USB assignment does not require special considerations. If one USB device is assigned to a virtual machine, access is mediated by *QEMU* via the USB device file. The access permissions on the USB device file are restricted by *libvirtd* modifying the device file ownership and its SELinux label such that only the virtual machine which is given access to the device is able to access the device file.

WARNING: Neither the *libvirtd* management daemon, nor the KVM functionality implement any constraints in which USB devices can be assigned to a virtual machine. A virtual machine administrator may assign any device listed in the `lsusb` output to a virtual machine. This includes devices that are needed by the host system. Virtual machine administrators have the ability to disrupt the operation of the host. The evaluated configuration does not place any constraints on which USB devices are assigned to virtual machines. This gives the virtual machine administrator full flexibility over USB device assignment configuration. However, the administrator MUST be very careful about which USB devices are assigned.

Chapter 6

Monitoring, Logging & Audit

6.1 Reviewing the system configuration

It is RECOMMENDED that you review the system's configuration at regular intervals to verify if it still agrees with the evaluated configuration. This primarily concerns those processes that run with root privileges.

The permissions of the device files */dev/** MUST NOT be modified.

In particular, review settings in the following files and directories to ensure that the contents and permissions have not been modified:

```
/etc/audit/audit.rules
/etc/audit/auditd.conf
/etc/cron.{ weekly hourly daily monthly}
/etc/cron.allow
/etc/cron.d/*
/etc/cron.deny
/etc/crontab
/etc/group
/etc/gshadow
/etc/hosts
/etc/inittab
/etc/ipsec.conf
/etc/ipsec.d/*
/etc/ipsec.secrets
/etc/ld.so.conf
/etc/libvirt/*
/etc/localtime
/etc/login.defs
/etc/modprobe.conf
/etc/netlabel.rules
/etc/pam.d/*
/etc/passwd
/etc/rc.d/init.d/*
/etc/rc.d/init.d/auditd
/etc/securetty
/etc/security/opasswd
/etc/selinux/config
/etc/selinux/mls/contexts/
```

```
        /etc/selinux/mls/modules/
        /etc/selinux/mls/policy/
        /etc/selinux/mls/setrans.conf
        /etc/selinux/mls/seusers
        /etc/selinux/semanage.conf
        /etc/shadow
        /etc/ssh/sshd_config
        /etc/sysconfig/*
        /etc/sysctl.conf
        /etc/xinetd.conf
        /etc/xinetd.d/*
        /var/log/lastlog
        /var/run/faillock/*
        /var/spool/cron/root
```

Use the commands `lastlog` as well as `last` to detect unusual patterns of logins.

Also verify the output of the following commands (run as root):

```
        crontab -l
        find / \( -perm -4000 -o -perm -2000 \) -ls
        find / \( -type f -o -type d -o -type b \) -perm -0002 -ls

        find /bin /boot /etc /lib /sbin /usr \
                ! -type l \( ! -uid 0 -o -perm +022 \)
```

6.2 System logging and accounting

System log messages are stored in the */var/log/* directory tree in plain text format, most are logged through the *syslogd*(8) and *klogd*(8) programs, which MAY be configured via the */etc/syslog.conf* file.

The *logrotate*(8) utility, launched from */etc/cron.daily/logrotate*, starts a fresh log file every week or when they reach a maximum size and automatically removes or archives old log files. You MAY change the configuration files */etc/logrotate.conf* and */etc/logrotate.d/** as required.

In addition to the *syslog* messages, various other log files and status files are generated in */var/log* by other programs:

File	Source
audit	Default audit log storage
boot.msg	Messages from system startup
libvirt	Log maintained by libvirtd
lastlog	Last successful log in (see lastlog(8))
localmessages	Written by syslog
mail	Written by syslog, contains messages from the MTA (postfix)
messages	Written by syslog, contains messages from su and ssh
news/	syslog news entries (not used in the evaluated configuration)
secure	Security related messages (for example from PAM)
warn	Written by syslog
wtmp	Written by the PAM susbystem, see who(1)

Please see *syslog*(3), *syslog.conf*(5) and *syslogd*(8) man pages for details on syslog configuration.

The *ps*(1) command can be used to monitor the currently running processes. Using `ps faux` will show all currently running processes and threads.

6.3 Configuring the audit subsystem

The audit subsystem implements a central monitoring solution to keep track of security relevant events, such as changes and change attempts to security critical files.

This is accomplished through two separate mechanisms. All system calls are intercepted, and the kernel writes the parameters and return value to the audit log for those calls that are marked as security relevant in the filter configuration. In addition, some trusted programs contain audit-specific code to write audit trails of the actions they are requested to perform.

Please refer to the *auditd*(8), *auditd.conf*(5), and *auditctl*(8) man pages for more information.

Refer to section §4.19 "Remote audit configuration" for details about the configuration of remote auditing.

6.3.1 Intended usage of the audit subsystem

The operational mode of Base and MLS specify the auditing capabilities that a compliant system must support. The evaluated configuration described here is based on these requirements.

WARNING: Some of the protection profile requirements can conflict with your specific requirements for the system. For example, a MLS-compliant system MUST disable logins if the audit subsystem is not working. Please ensure that you are aware of the consequences if you enable auditing.

Base and MLS are designed for a multiuser system, with multiple unique users who maintain both shared and private resources. The auditing features are intended to support this mode of operation with a reliable trail of security-relevant operations. It is less useful for a pure application server with no interactive users.

Please be aware that the auditing subsystem will, when activated, cause some slowdown for applications on the server. The impact depends on what the application is doing and how the audit subsystem is configured. As a rule of thumb, applications that operate on a large number of separate files are most affected, and CPU-bound programs should not be measurably affected. You will need to balance the performance requirements against your security needs when deciding if and how you want to use auditing.

6.3.2 Selecting the events to be audited

You MAY make changes to the set of system calls and events that are to be audited. Base and MLS require that the system has the *capability* to audit security relevant events, but it is up to you to choose how you want to use these capabilities. It is acceptable to turn off system call auditing completely even in an evaluated configuration, for example on a pure application server with no interactive users on the system.

The audit package provides several suggested audit configuration files, for example the */usr/share/doc/audit-*/capp.rules* file for Base systems, and the *lspp.rules* file (in the same location) for MLS systems. They contain a suggested setup for a typical multiuser system, all access to security relevant files is audited, along with other security relevant events such as system reconfiguration. You MAY copy one of the sample rules files to */etc/audit/audit.rules* and modify the configuration according to your local requirements, including the option of using an empty audit rules file to disable auditing if not required.

The man page *audit.rules*(7) provides a number of tips as well as auditing strategies.

When using CUPS in MLS mode, it is RECOMMENDED to configure an audit rule to monitor changes to the printer device MLS level, for example:

```
-w /dev/lp1 -k Printdevice
-a exit,possible -S setxattr
```

You MAY selectively disable and enable auditing for specific events or users as required by modifying the *audit.rules* file.

It is RECOMMENDED that you monitor use of the *semodule*(8) tool to keep track of administrative changes to optional security policy modules:

```
-w /usr/sbin/semodule
```

It is RECOMMENDED that you always reconfigure the audit system by modifying the */etc/audit/audit.rules* file and then running the following command to reload the audit rules:

```
# as role "auditadm_r"
auditctl -R /etc/audit/audit.rules
```

This procedure ensures that the state of the audit system always matches the content of the */etc/audit/audit.rules* file. You SHOULD NOT manually add and remove audit rules and watches on the command line as those changes are not persistent.

Note that reloading audit rules involves initially deleting all audit rules, and for a short time the system will be operating with no or only a partial set of audit rules. It is RECOMMENDED to make changes to the audit rules when no users are logged in on the system, for example by using single user mode or a reboot to activate the changes.

Please refer to the *auditctl*(8) man page for more details.

Setting SELinux contexts through library functions such as *setexeccon*(3), or equivalently by writing information to the */proc/self/attr/{current,exec,fscreate,sockcreate}* files, will generate audit records when enabled in the SELinux policy by *auditallow* rules. Please refer to section §4.17.2 "MLS mode specific policy module" of this guide for more information about configuring this policy. Note that the audit records will always report success when the open/write operation to the pseudofile was successful, even if the context that was written there is invalid. In the case of an invalid context, the following operation (such as an *exec*(2) system call in the case of *setexeccon*) will fail, and will generate its own audit record if configured to do so for that operation.

6.3.3 Reading and searching the audit records

Use the *ausearch*(8) tool to retrieve information from the audit logs. The information available for retrieval depends on the active filter configuration. If you modify the filter configuration, it is RECOMMENDED keeping a datestamped copy of the applicable configuration with the log files for future reference.

For example:

```
# search for events with a specific login UID
ausearch -ul jdoe

# search for events by process ID
ausearch -p 4690
```

Please refer to the *ausearch*(8) man page for more details. In addition, the *audit.rules*(7) man page contains additional hints on implementing effective audit trail searches.

For some system calls on some platforms, the system call arguments in the audit record can be slightly different than you may expect from the program source code due to modifications to the arguments in the C library or in kernel wrapper functions. For example, the *mq_open*(3) glibc library function strips the leading '/' character from the path argument before passing it to the *mq_open*(2) system call, leading to a one character difference in the audit record data. Similarly, some system calls such as *semctl*(2), *getxattr*(2), and *mknodat*(2) can have additional internal flags automatically added to the flag argument. These minor modifications do not change the security relevant information in the audit record.

Of course, you can use other tools such as plain *grep*(1) or scripting languages such as *awk*(1), *python*(1) or *perl*(1) to further analyze the text audit log file or output generated by the low-level *ausearch* tool.

6.3.4 Starting and stopping the audit subsystem

If the audit daemon is terminated, no audit events are saved until it is restarted. To avoid lost audit records when you have modified the filter configuration, you MUST use the command `service auditd reload` to re-load the filters.

You MUST NOT use the *KILL* signal (-9) to stop the audit daemon, doing so would prevent it from cleanly shutting down.

It is RECOMMENDED that you add the kernel parameter `audit=1` to your boot loader configuration file to ensure that all processes, including those launched before the *auditd* service, are properly attached to the audit subsystem. Please refer to the documentation of your boot loader and section §3.9 "Configuring the boot loader" of this document for more details.

6.3.5 Storage of audit records

The default audit configuration stores audit records in the */var/log/audit/audit.log* file. This is configured in the */etc/audit/auditd.conf* file. You MAY change the *auditd.conf* file to suit your local requirements.

It is RECOMMENDED that you configure the audit daemon settings appropriately for your local requirements, for example by changing the log file retention policy to never delete old audit logs with the following setting in the */etc/audit/auditd.conf* file:

```
max_log_file_action = KEEP_LOGS
```

The most important settings concern handling situations where the audit system is at risk of losing audit information, such as due to lack of disk space or other error conditions. You MAY choose actions appropriate for your environment, such as switching to single user mode (action `single`) or shutting down the system (action `halt`) to prevent auditable actions when the audit records cannot be stored.

Warning: Switching to single user mode does not automatically kill all user processes when using the system default procedure. You MAY kill processes of users by using `killall -u`. Please note that system services SHOULD NOT be terminated. Depending on your local policy, you MAY need to shut down KVM guests. As these KVM guests act like normal processes on the Linux host system, the same commands to terminate these processes may be used.

Halting the system is RECOMMENDED and most certain way to ensure all user processes are stopped. The following settings are RECOMMENDED in the */etc/audit/auditd.conf* file if a fail-secure audit system is required:

```
admin_space_left_action = SINGLE
disk_full_action = HALT
disk_error_action = HALT
```

It is RECOMMENDED that you configure appropriate disk space thresholds and notification methods to receive an advance warning when the space for audit records is running low.

It is RECOMMENDED that you use a dedicated partition for the */var/log/audit/* directory to ensure that *auditd* has full control over the disk space usage with no other processes interfering.

Please refer to the *auditd.conf*(5) man page for more information about the storage and handling of audit records.

6.3.6 Reliability of audit data

You MAY choose an appropriate balance between availability of the system and secure failure mode in case of audit system malfunctions based on your local requirements.

You MAY configure the system to cease all processing immediately in case of critical errors in the audit system. When such an error is detected, the system will then immediately enter "panic" mode and will need to be manually rebooted. To use this mode, add the following line to the */etc/audit/audit.rules* file:

```
-f 2
```

Please refer to the *auditctl*(8) man page for more information about the failure handling modes.

You MAY edit the */etc/libaudit.conf* file to configure the desired action for applications that cannot communicate with the audit system. Please refer to the *get_auditfail_action*(3) man page for more information.

auditd writes audit records using the normal Linux filesystem buffering, which means that information can be lost in a crash because it has not been written to the physical disk yet. Configuration options control how *auditd* handles disk writes and allow the administrator to choose an appropriate balance between performance and reliability.

Any applications that read the records while the system is running will always get the most current data out of the buffer cache, even if it has not yet been committed to disk, so the buffering settings do not affect normal operation.

The default setting is `flush = DATA`, ensuring that record data is written to disk, but metadata such as the last file time might be inconsistent.

The highest performance mode is `flush = none`, but be aware that this can cause loss of audit records in the event of a system crash.

If you want to ensure that auditd always forces a disk write for each record, you MAY set the `flush = SYNC` option in */etc/audit/auditd.conf*, but be aware that this will result in significantly reduced performance and high strain on the disk.

A compromise between crash reliability and performance is to ensure a disk sync after writing a specific number of records to provide an upper limit for the number of records lost in a crash. For this, use a combination of `flush = INCREMENTAL` and a numeric setting for the `freq` parameter, for example:

```
flush = INCREMENTAL
freq = 100
```

The audit record files are *not* protected against a malicious administrator, and are not intended for an environment where the administrators are not trustworthy.

6.4 System configuration variables in */etc/sysconfig*

The system uses various files in */etc/sysconfig* to configure the system. Most files in this directory tree contain variable definitions in the form of shell variables that are either read by the rc scripts at system boot time or are evaluated by other commands at runtime. Note that changes will not take effect until the affected service is restarted or the system is rebooted.

Chapter 7

Security guidelines for users

7.1 System Documentation

The system provides a large amount of online documentation, usually in text format. Use the `man` program to read entries in the online manual, for example:

```
man ls
man man
```

to read information about the `ls` and `man` commands respectively. You can search for keywords in the online manual with the *apropos*(1) utility, for example:

```
apropos password
```

When this guide refers to manual pages, it uses the syntax ENTRY(SECTION), for example *ls*(1). Usually you do not need to provide the section number, but if there are several entries in different sections, you can use the optional `-S` switch and pick a specific one.

Some programs provide additional information GNU 'texinfo' format, use the `info` program to read it, for example:

```
info diff
```

Additional information, sorted by software package, may be found in the */usr/share/doc/*/* directories. Use the *less*(1) pager to read it, for example:

```
/usr/share/doc/pam-*/txts/README*
```

Many programs also support a `--help`, `-?` or `-h` switch you can use to get a usage summary of supported command-line parameters.

Note that this Configuration Guide has precedence over other documents in case of conflicting recommendations.

7.2 Authentication

You MUST authenticate (prove your identity) before being permitted to use the system. When the administrator created your user account, he or she will have assigned a user name and default password, and provided that information for you along with instructions how to access the system.

Logging in to the system will usually be done using the Secure Shell (SSH) protocol, alternatively a serial terminal can be used. Use the `ssh` command to connect to the system unless instructed otherwise by the administrator, for example:

```
        ssh jdoe@172.16.0.1
```

In case the system administrator has assigned multiple roles for your use, you can select the desired role by appending it to the username separated with a slash, for example:

```
        ssh jdoe/user_r@172.16.0.1
```

The *ssh*(1) manual page provides more information on available options. If you need to transfer files between systems, use the *scp*(1) or *sftp*(1) tools.

If this is the first time you are connecting to the target system, you will be prompted if you want to accept the host key. If the administrator has provided a key fingerprint for comparison, verify that they match, otherwise type `yes` to continue. You MUST immediately change your initially assigned password with the *passwd*(1) utility.

You MUST NOT under any circumstances attempt to log in from an insecure device, such as a public terminal or a computer belonging to a friend. Even if the *person* owning the computer is trustworthy, the *computer* might not be due to having been infected with malicious code. Always remember that the device you are typing your password into has the ability to save and re-use your authentication information, so you are in effect giving the computer you are using the right to do any and all actions in your name. Insecure handling of authentication information is the leading cause for exploits of otherwise secure systems, and SSH can only protect the information during transit, and offers no protection at all against an insecure end point.

When you log out from the system and leave the device you have used for access (such as a terminal or a workstation with terminal emulation), you MUST ensure that you have not left information on the screen or within an internal buffer that should not be accessible to another user. You should be aware that some terminals also store information not displayed on the terminal (such as passwords, or the contents of a scrollback buffer). Nevertheless this information can be extracted by the next user unless the terminal buffer has been cleared. Safe options include completely shutting down the client software used for access, powering down a hardware terminal, or clearing the scrollback buffer by switching among virtual terminals in addition to clearing the visible screen area.

If you ever forget your password, contact your administrator who will be able to assign a new password.

You MAY use the *chsh*(1) and *chfn*(1) programs to update your login shell and personal information if necessary. Not all settings can be changed this way, contact your administrator if you need to change settings that require additional privileges.

7.3 Password policy

All users, including the administrators, MUST ensure that their authentication passwords are strong (hard to guess) and handled with appropriate security precautions. The password policy described here is designed to satisfy the requirements of the evaluated configuration. If your organization already has a password policy defined, your administrator MAY refer you to that policy if it is equivalently strong.

You MUST change the initial password set by the administrator when you first log into the system. You MUST select your own password in accordance with the rules defined here. You MUST also change the password if the administrator has set a new password, for example if you have forgotten your password and requested the administrator to reset the password.

Use the *passwd*(1) program to change passwords. It will first prompt you for your old password to confirm your identity, then for the new password. You need to enter the new password twice, to catch mistyped passwords.

The *passwd*(1) program will automatically perform some checks on your new password to help ensure that it is not easily guessable, but you MUST nevertheless follow the requirements in this chapter.

Note that the administrators MUST also ensure that their own passwords comply with this password policy, even in cases where the automatic checking is not being done, such as when first installing the system.

- Your password MUST be a minimum of 8 characters in length. More than 8 characters MAY be used (it is RECOMMENDED to use more than 8, best is to use passphrases), and all characters are significant.

- Combine characters from different character classes to construct a sufficiently strong password, using either 8 total characters containing at least one character from each class, or alternatively 12 total characters chosen from any three of the classes. The character classes are defined as follows:

  ```
  Lowercase letters: abcdefghijklmnopqrstuvwxyz
  Uppercase letters: ABCDEFGHIJKLMNOPQRSTUVWXYZ
  Digits:            0123456789
  Punctuation:       !"#$%&’()*+,-./:;<=>?[\]^_`{|}~
  ```

  Note that non-7-bit ASCII characters MAY be used for passwords.

- You MUST NOT base the password on a dictionary word, your real name, login name, or other personal details (such as dates, names of relatives or pets), or names of real people or fictional characters.

- Instead of a password, you MAY use a passphrase consisting of multiple unrelated words (at least three) joined with random punctuation characters. Such a passphrase MUST have a length of at least 16 characters. (This corresponds to automatically generated pass phrases constructed by choosing 3 words from a 4096 word dictionary and adding two punctuation characters from a set of 8, equivalent to 42 bits of entropy.)

- You MUST NOT use a simple alphabetic string, palindrome or combinations of adjacent keyboard keys.

- When you choose a new password, it MUST NOT be a simple variation or permutation of a previously used one.

- You MUST NOT write the password on paper or store it on electronic devices in unprotected form. Storage in a secure location (such as an envelope in a safety deposit box, or encrypted storage on an electronic device) MAY be acceptable, contact your administrator first to ensure that the protection is strong enough to make password recovery infeasible for the types of attackers the system is intended to protect against.

- The password is for you and you only. A password is like a toothbrush - you do not want to share it with anybody, even your best friend. You MUST NOT disclose your password to anybody else, or permit anybody else to use the system using your identity.

  Note that administrators will never ask you for your password, since they do not need it even if they are required to modify settings affecting your user account.

- You MUST NOT use the same password for access to any systems under external administration, including Internet sites. You MAY however use the same password for accounts on multiple machines within one administrative unit, as long as they are all of an equivalent security level and under the control of the same administrators.

- You MUST inform the administrator and select a new password if you have reason to believe that your password was accidentally disclosed to a third party.

- If the system notifies you that your password will expire soon or has expired, choose a new one as instructed. Contact your administrator in case of difficulty.

A RECOMMENDED method of generating passwords that fits these criteria while still being easy to memorize is to base it on letters of words in a sentence (NOT a famous quotation), including capitalization and punctuation and one or two variations. Example:

```
"Ask not for whom the bell tolls."
=> An4wtbt.

"Password 'P'9tw;ciSd' too weak; contained in RHEL documentation"
=> P'9tw;ciRd
```

7.4 SSH key-based authentication

You MAY use the SSH key-based authentication documented in *sshd*(8) section "AUTHORIZED_KEYS FILE FORMAT". Before the SSH key-based authentication can be used, you must generate either an RSA or DSA key pair using the *ssh-keygen*(1) utility.

As only the `ssh-keygen` utility provided with the TOE was subject to the security assessment, including the proper key generation support, it is strongly RECOMMENDED that you use this tool from the TOE.

You MUST generate either RSA or DSA key pairs for SSHv2 using the `-t rsa` or `-t dsa` command line switch.

The `ssh-keygen` utility allows you to specify the key size for RSA with the default of 2048 bits. If you select a different key size, you MUST use either 1024 bits or 3072 bits.

You MUST keep the private key part stored in *~/.ssh/id_dsa* or *~/.ssh/id_rsa* inaccessible to any other user. This file must be treated similarly to a password. It is strongly RECOMMENDED that you protect that key with a passphrase using `ssh-keygen`.

The following command line is an example that generates a DSA key with 1024 bit key size:

```
ssh-keygen -t dsa -C "John Doe's key"
```

The command asks you for a passphrase where you SHOULD provide a strong passphrase.

After the generation of the key pair, you MAY copy the file *~/.ssh/id_dsa.pub* or *~/.ssh/id_rsa.pub* to your server system and append it to the file *~/.ssh/authorized_keys*. Create that file if it does not exist and ensure that its permission prevents others from accessing this file. More information can be found in *sshd*(8) section "AUTHORIZED_KEYS FILE FORMAT".

In case you fail to meet the above mentioned requirements, your account protection may be weakened. This can be considered similar to choosing a weak password or fail to keep the password confidential.

Please note that using the key-based authentication is not subject to the account locking mechanism enforced for passwords.

7.5 Access control for files and directories

Linux is a multiuser operating system, and it is essential that the system can enforce confidentiality and integrity of user data. For this purpose, the operating system implements access control policies that provide rules for reading and writing data.

Note that the administrators (root) are able to override these permissions and access all files on the system. Use of encryption is RECOMMENDED for additional protection of sensitive data.

7.5.1 Discretionary Access Control

You can control which other users will be able to read or modify your files by setting the Unix permission bits and user/group IDs, or (if more precise control is needed) by using POSIX-style access control lists (ACLs). This is referred to as discretionary access control (DAC).

The 'umask' setting controls the permissions of newly created files and directories and specifies the access bits that will be *removed* from new objects. Ensure that the setting is appropriate, and never grant write access to others by default. The umask MUST include at least the 002 bit (no write access for others), and the RECOMMENDED setting is 027 (read-only and execute access for the group, no access at all for others). The default configuration is even more strict as it sets 077 (accessible to the owner only).

Do not set up world-writable areas in the filesystem - if you want to share files in a controlled manner with a fixed group of other users (such as a project group), please contact your administrator and request the creation of a user group for that purpose.

Programs can be configured to run with the access rights of the program file's owner and/or group instead of the rights of the calling user. This is the SUID/SGID mechanism, which utilities such as *passwd*(1) use to be able to access security-critical files. You could also create your own SUID/SGID programs via *chmod*(1), but DO NOT do that unless you fully understand the security implications - you would be giving away *your* access privileges to whoever launches the SUID program. Please refer to the "Secure Programming HOWTO" in the unlikely case that you need to create such a program, there you will find explanations of the many aspects that must be considered, such as the risk of unintended shell escapes, buffer overflows, resource exhaustion attacks and many other factors. Note that SUID root programs MUST NOT be added to the evaluated configuration, the only permitted use of the SUID bit is for setting non-root user IDs.

7.5.2 Multilevel mandatory access control (MLS mode only)

The system can enforce additional restrictions on operations. When the system is in MLS mode, it enforces mandatory access control (MAC) to ensure that all data and user processes are labeled and that information flow is possible only according to rules based on these labels. The rules are "mandatory" since you cannot voluntarily give read access to other users for data if that would violate the information flow rules.

Users can be cleared to operate at multiple different MLS levels, but each interactive session has a single effective level. A MLS level consists of a hierarchical component (for example "s0" or "s4") and zero or more categories separated by commas. Contiguous categories can be abbreviated with the first and last category separated by a period ("."). A sample MLS level specification is "s2:c2,c5.c7,c9", equivalent to "s2:c2,c5,c6,c7,c9".

You MAY use the following methods to select an MLS level and categories for your interactive session:

- At a login prompt (for example on a serial terminal), you will be prompted interactively for the desired role and MLS level.
- Run `newrole -l` to launch a new shell running at a different level from within the current session, for example:

  ```
  newrole -l s2:c1,c3.c5
  ```

  This functionality is restricted to secure terminal types listed in the */etc/selinux/mls/contexts/securetty_types* file, and is not supported for pseudoterminals as used in *ssh* sessions.
- When labeled networking is active, you MUST specify the correct port number for the label aware sshd instance when this number is different from 22, for example:

  ```
  ssh -p 222 user@rhel5.example.com
  ```

When labeled networking is disabled, *ssh* supports selecting a role and level when logging in, using the *user/role/mlslevel@host* syntax, for example:

```
ssh jdoe/staff_r/s2:c0,c1@rhel5.example.com
```

Administrators MAY use the *chcon -l*(1) command to change the MLS labels for objects, this is NOT permitted for regular users. Please contact an administrator if you believe that data objects are incorrectly labeled.

The *chcat*(8) program is NOT supported in the evaluated configuration and is disabled. It is intended for use with the Multi-Category System (MCS) security policy which is distinct from the MLS policy. The MCS policy is beyond the scope of this guide and evaluation.

7.5.3 Role-based access control (MLS mode only)

In addition, in MLS mode, the system supports role-based access control (RBAC) to further restrict access according to administrator-defined rules. Permissions are based on roles assigned to users, and you may have the option of choosing from several roles when logging in to the system, or by using the *newrole -r ROLE* command. You are only able to choose a role from the set of roles that the system administrator has assigned for your use, and you will use a default role when you do not select one explicitly. Please contact your system administrator for further information about which roles are available and what the corresponding access rights and restrictions are.

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

Permissions to perform actions are delegated to specific roles. In addition the system supports types that can be associated with objects and domains that can be associated with processes. Roles are defined by the domains they have access to. A predefined policy file, which is part of the system configuration, defines the rules between domains and types.

You MAY use the *chcon -t* command to change the SELinux type of objects which can grant permission to specific roles to access the object. You are limited to selecting from a small number of allowed types for this purpose, the tool will reject attempts to change to restricted types. Your system administrator will explain the types available for this purpose. For example, in the default configuration, you MAY switch between the *user_home_t* and *user_home_ssh_t* types:

```
chcon -t user_home_t FILE
chcon -t user_home_ssh_t FILE
```

(This example is not useful since the default configuration does not define user roles with granular permissions. Additional types will be available when your system administrator has defined additional roles and the corresponding types.)

7.5.4 General access control

Access is permitted only if all policies (DAC, MAC, and RBAC) agree in permitting the access.

Always remember that **you** are responsible for the security of the data you create and use. Choose permissions that match the protection goals appropriate for the content, and that correspond to your organization's security policy. Access to confidential data MUST be on a need-to-know basis, do not make data world-readable unless the information is intended to be public.

Whenever you start a program or script, it will execute with your access rights. This implies that a malicious program would be able to read and modify all files that you have access to. Never execute any code that you have received from untrustworthy sources, and do not run commands that you do not understand. Be aware that manipulations to the environment a program is run in can also cause security flaws, such as leaking sensitive information. Do not use the shell variables LD_LIBRARY_PATH or LD_PRELOAD that modify the shared library configuration used by dynamically linked programs unless the specific settings are approved by the administrator or your organizational policies.

Please refer to the *chmod*(1), *umask*(2), *chown*(1), *chgrp*(1), *acl*(5), *getfacl*(1), and *setfacl*(1) manual pages for information, or any of the many available books covering Linux security (cf. Appendix 'Literature'), or ask your system administrator for advice.

7.6 Data import / export

The system comes with various tools to archive data (*tar*, *cpio*). If ACLs or file labels are used, then only *tar* MUST be used to handle the files and directories as the other commands do not support ACLs. The options `-acl` for ACLs as well as `-selinux` for MLS lables must be used with *tar*.

Please see the *tar*(1) man page for more information.

7.7 Screen saver

The system is provided with the possibility to lock your terminal. To unlock the terminal, you MUST provide your password.

The locking is established using the `screen` application. Depending on the system configuration, `screen` MAY already be started during login. If the `screen` application is not started, you may start it manually.

The `screen` application allows the following two types of screen locking:

- Automated locking of the screen after a period of inactivity on the terminal defined by a timeout in either */etc/screenrc* or *~/.screenrc* using the *lockscreen* configuration value.
- Manual locking by executing the `C-a C-x` screen key binding combination.

You MAY change the timeout value for locking the session in *~/.screenrc* with the value for *lockscreen*. Note that the administrator MAY disable the ability to use the *~/.screenrc* configuration file.

If `screen` is not invoked automatically during startup, you MAY enter the following line to *~/.bash_profile*.

```
exec screen
```

7.8 Buffer overflow hardening

The system provides hardening techniques for applications and shared libraries that raises the bar for exploiting buffer overflows significantly. The following list enumerates the hardening techniques which are available including the instructions on how to enable them.

- The memory hosting the process stack is marked as not hosting executable code. This means that the CPU will not execute instructions that are stored on the stack memory. To enable the functionality globally, file */proc/sys/kernel/exec-shield* must contain either a 1 or 2. Either the value is written into that file directly or the `sysctl` command is used.

In addition, the application that tries to be covered by the no-execute stack must obey the following rules. During standard compilation of applications, the stack execution protection is enabled. To ensure the presence of the PT_GNU_STACK ELF program header entry and the absence of the PF_X bit in the p_flags ELF header flags (i.e. the flags that influence the disabling of the non-executable stack support for a particular application), the following considerations must be applied by a programmer as any of the following operations disable the stack execution protection:

- The following linker option must **not** be used: *"-z execstack"* (gcc: *"-Wl,-z,execstack"*).
- The following assembler option must **not** be used: *"–execstack"* (gcc: *"-Wa,–execstack"*).
- Modifications of an ELF program header entry in an already compiled binary which change the PT_GNU_STACK and PF_X flags (like using the `execstack`(8) application) must **not** be performed.
- The application or library code must not contain trampolines such as nested functions pushed onto the stack which passed as pointers to functions as this would also enable the stack execution support.

- The kernel applies the concept of address space randomization to applications. This implies that the offset addresses of symbols in applications and shared libraries are different for each application invocation. To enable the functionality globally, file */proc/sys/kernel/exec-shield-randomize* must contain a 1. Again, this file contents can be changed by either writing directly into it or by using the `sysctl` command.

- During load time of an application and its depending shared libraries, the tables holding the offset information for objects and functions can be marked read-only. This functionality can be enabled on an application or shared library base as it is enforced by the program loader rather than the kernel. This mechanism is called RELRO/PIE and is usable in two different varieties. The following sections outline the different flavors of RELRO/PIE and outline the functional difference.

7.8.1 Full RELRO/PIE

By enabling full RELRO/PIE for an application or shared library, the data pointed to by the following ELF sections are loaded into read-only memory before the application gains control over the process: .tdata, .preinit_array, .init_array, .fini_array, .ctors, .dtors, .jcr, .data.rel.ro, .dynamic, .got including .got.plt.

Full RELRO/PIE implies that the offsets of all symbols of an application and all its libraries are resolved before the application starts executing. A performance penalty applies compared to uncovered applications as the concept of lazy binding which is implemented by the loader cannot be used any more.

In order to enable full RELRO, the ELF program header entry of PT_GNU_RELRO must be set such that it covers the proper elf sections. To achieve that, the application must be linked with the provided RHEL linker using the linker options of *"-z relro -z now"* (using the GCC compiler using the compiler options of *"-Wl,-z,relro,-z,now"* can be used which are passed to the linker). In addition, an application must be compiled as PIE with the gcc option of "-fPIE". Contrary, a shared library must be compiled as PIC with the gcc option of "-fPIC".

The ELF header sections listed above are set read-only using the mprotect system call by the loader before the application gains control. When exploiting buffer overruns, the attacker cannot modify information in those memory sections. These sections store offset tables required for the dynamic linking mechanism and, if abused, allow attackers to modify the jump addresses of object accesses. Full protection against this type of attack can only be achieved if the application and all depending shared libraries are compiled linked with full protection enabled. When at least one shared library the application depends on or the application itself is compiled and linked with partial RELRO/PIE protection as discussed in the following section, only partial protection against this type of attack is available for the given application. Also, if at least one shared library the application loads does not have any RELRO protection enabled, the entire runtime environment of the application cannot considered to have RELRO protection any more as the logic provided by the uncovered library may provide an attacker a window into cracking the entire application.

7.8.2 Partial RELRO/PIE

The partial RELRO/PIE protection is almost identical to the full RELRO/PIE protection. The only difference is that the data covered with the ELF section of .got.plt (i.e. the Procedure Loading Table – PLT) is left in read/write memory. The user-visible difference is that lazy binding is still available. Therefore, the linker can employ the concept of lazy binding. On the other hand, the function offsets maintained with the PLT are not protected against modification by determined attackers.

In order to enable partial RELRO, the ELF program header entry of PT_GNU_RELRO must be set such that it covers the proper elf sections. To achieve that, the application must be linked with the provided RHEL linker using the linker options of *"-z relro"* (using the GCC compiler using the compiler options of *"-Wl,-z,relro"* can be used which are passed to the linker). In addition, an application must be compiled as PIE with the gcc option of "-fPIE". Contrary, a shared library must be compiled as PIC with the gcc option of "-fPIC".

Chapter 8

Appendix

8.1 Online Documentation

If there are conflicting recommendations in this guide and in one of the sources listed here, the Configuration Guide has precedence concerning the evaluated configuration.

RHEL6 Guidance: *http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/index.html*

David A. Wheeler, "Secure Programming for Linux and Unix HOWTO", *http://tldp.org/HOWTO/Secure-Programs-HOWTO/*

Kevin Fenzi, Dave Wreski, "Linux Security HOWTO", *http://tldp.org/HOWTO/Security-HOWTO/*

8.2 Literature

Frank Mayer, Karl MacMillan, David Caplan, "SELinux by Example: Using Security Enhanced Linux", Prentice Hall 2006, ISBN 0131963694

Ellen Siever, Stephen Spainhour, Stephen Figgins, & Jessica P. Hekman, "Linux in a Nutshell, 6rd Edition", O'Reilly 2009, ISBN 978-0596154486

W. Richard Stevens, "Advanced Programming in the UNIX(R) Environment", Addison-Wesley 1992, ISBN 0201563177