



EAL4 Evaluated Configuration Guide for Red Hat Enterprise Linux with KVM support on IBM hardware

December 14, 2011; v1.4

atsec is a trademark of atsec GmbH

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

Java and all Java-based products are trademarks of Sun Microsystems, Inc., in the United States, other countries, or both.

Linux is a registered trademark of Linus Torvalds.

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

All other trademarks are property of their respective owners.

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

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

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"

Changes Copyright (c) 2004, 2005, 2006, 2007 by atsec corp., Hewlett-Packard Company or its wholly owned subsidiaries.

Changes Copyright (c) 2005, 2007, 2008 by atsec corp., and Silicon Graphics Incorporated or its wholly owned subsidiaries.

Changes Copyright (c) 2009 by atsec corp., and DELL Incorporated or its wholly owned subsidiaries.

Changes Copyright (c) 2011 by atsec corp., and Red Hat Incorporated or its wholly owned subsidiaries.

Contents

1	Introduction	1
1.1	Purpose of this document	1
1.2	How to use this document	1
1.3	Requirements and assumptions	2
1.3.1	What is a CC compliant system?	2
1.3.2	Hardware requirements	2
1.3.3	Requirements for the system's environment	3
1.3.4	Requirements for connectivity	3
1.3.5	Requirements for administrators	3
1.3.6	Requirements for the system's users	4
2	Installation	5
2.1	Supported hardware	5
2.2	Selection of install options and packages	5
2.2.1	Prerequisites for installation	6
2.2.2	Preparing for installation	6
2.2.3	Customizing the installation	8
2.2.4	Kickstart	9
2.2.5	Pre-install configuration	10
2.2.6	Partitioning	11
2.2.7	Post-install configuration	11
3	Secure initial system configuration	15
3.1	Add and remove packages	15
3.2	Creating additional user accounts for administrators	15
3.3	Installing required updates	16
3.4	Automated configuration of the system	16
3.5	Setting up the Cups printing system	16
3.6	Setting up Postfix	17
3.7	Configuring the boot loader	17
3.7.1	GRUB boot loader configuration	17
4	System operation	19
4.1	System startup, shutdown and crash recovery	19
4.2	Backup and restore	19
4.3	Gaining administrative access	20
4.4	Installation of additional software	20
4.4.1	Supported software architectures	20
4.4.2	Security requirements for additional software	21
4.5	Scheduling processes using cron	22
4.6	Mounting filesystems	23
4.7	Managing user accounts	24
4.7.1	Creating users	24

4.7.2	Changing user passwords	25
4.7.3	SSH key-based authentication	25
4.7.4	Changing user properties	25
4.7.5	Locking and unlocking user accounts	25
4.7.6	Removing users	26
4.7.7	Defining administrative accounts	26
4.8	Using serial terminals	27
4.9	Managing data objects	27
4.9.1	Revoking access	27
4.9.2	SYSV shared memory and IPC objects	27
4.9.3	Posix Message Queues	27
4.10	Configuring object access rights	28
4.11	Setting the system time and date	28
4.12	Firewall configuration	28
4.13	Screen saver configuration	29
4.14	OpenSSH configuration	29
5	Kernel-based virtual machine (KVM) management	31
5.1	Hardware configuration	31
5.2	Kernel-based Virtual Machine (KVM) configuration	32
5.2.1	Virtual machines started outside libvirt	33
5.2.2	System versus session instances of libvirt	33
5.3	Separation of virtual machines	33
5.3.1	Unprivileged user and group IDs	33
5.3.2	SELinux dynamic labeling	34
5.3.3	SELinux static labeling	34
5.4	Networking considerations	35
5.4.1	Packet filtering	35
5.5	Disk space management	35
6	Monitoring, Logging & Audit	37
6.1	Reviewing the system configuration	37
6.2	System logging and accounting	38
6.3	Configuring the audit subsystem	38
6.3.1	Intended usage of the audit subsystem	39
6.3.2	Selecting the events to be audited	39
6.3.3	Reading and searching the audit records	40
6.3.4	Starting and stopping the audit subsystem	40
6.3.5	Storage of audit records	40
6.3.6	Reliability of audit data	41
6.4	System configuration variables in <i>/etc/sysconfig</i>	42
7	Security guidelines for users	43
7.1	System Documentation	43
7.2	Authentication	44
7.3	Password policy	44
7.4	Access control for files and directories	46
7.4.1	Discretionary Access Control	46
7.4.2	General access control	46
7.5	Data import / export	47
7.6	Screen saver	47
8	Appendix	49
8.1	Online Documentation	49
8.2	Literature	49

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 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.

- IBM System x based on x86 64bit Intel Xeon processors: 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

These systems must be installed with the 64-bit version of RHEL 5.6.

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.

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.

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.

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).

1.3.5 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 (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.

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. 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 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`.

1.3.6 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.
- 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 5.6 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.
- All printers supported by CUPS that are attached to the system using a parallel port or USB connection are allowed. 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 §4.8 "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 5.6 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 SHA-256 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:

- `cc-eal4-config-rhel56` for the noarch architecture
- `openssh-4.3p2-72.el5_6.3` for x86_64 architecture
- `openssh-clients-4.3p2-72.el5_6.3` for x86_64 architecture
- `openssh-server-4.3p2-72.el5_6.3` for x86_64 architecture
- `libvirt-0.8.2-15.el5_6.3` for x86_64 architecture
- `libvirt-python-0.8.2-15.el5_6.3` for x86_64 architecture
- `selinux-policy-2.4.6-300.el5_6.1` noarch package

- selinux-policy-devel-2.4.6-300.el5_6.1 noarch package
- selinux-policy-mls-2.4.6-300.el5_6.1 noarch package
- selinux-policy-targeted-2.4.6-300.el5_6.1 noarch package

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-rhel56* RPM, the unpacked kickstart file (contained within the *cc-eal4-config-rhel56* RPM), and a specific set of RPM packages containing post-RHEL5.6 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/37017186.txt
```

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

```
rpm --import 37017186.txt
rpm --checksig cc-eal4-config-rhel56-*.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-rhel56* RPM into a temporary directory:

```
mkdir cc-inst
cd cc-inst
rpm2cpio ../cc-eal4-config-rhel56-*.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-rhel56-*/
    GPL.txt
    RHEL56-EAL4-Configuration-Guide.*

# the kickstart configuration used to automate the installation
./usr/share/cc/kickstart/
    ks-x86_64.cfg

# 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 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 §4.12 "Firewall configuration" of this guide for more information.

default set of optional packages

You MAY delete packages from the optional packages list

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 §2.2.6 "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.

2.2.4 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/#RHEL5>.

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.

The first boot parameter is the name of the booted kernel image, this is always `linux` for installation.

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
```

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=rhel5.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.

Examples:

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

# interactive network install, get IP address via DHCP
linux ks=http://example.com/rhel5/ks-x86_64.cfg

# noninteractive network install (all on a single line)
linux ip=172.16.2.5 netmask=255.255.255.0 gateway=172.16.2.1
    dns=172.16.2.1
    ks=http://example.com/rhel5/ks-x86_64.cfg
    method=cdrom:
    hostname=rhel5.example.com
    instdisk=sda
```

2.2.5 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] ?
```

```
Hostname (fully qualified) [rhel5.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 noninteractively.
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.

2.2.6 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** 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:

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

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

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

2.2.7 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 ***
```

Please verify the system time and date:

Local time: Wed Jan 26 00:28:16 CDT 2011

Universal time (UTC): Wed Jan 26 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-rhel56-0.10-1.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 [ftp://ftp.redhat.com/pub/redhat/linux/eal/EAL4_RHEL5/] ? !

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 [ftp://ftp.redhat.com/pub/redhat/linux/eal/EAL4_RHEL5/] ? /mnt/rpms/

[...]

Preparing...

[100%]

[...]

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

Uid [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.

Chapter 3

Secure initial system configuration

After the initial installation using the procedure described in the previous section, the operating system is in the evaluated configuration.

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 Add and remove packages

The kickstart automated install uses a default package selection that contains all packages required for the evaluated configuration. It also installs several optional packages that you MAY remove once the installation is complete.

For obtaining the list of packages that MAY be removed, please read the kickstart configuration file mentioned in §2.2.4 "Kickstart" and search for:

```
## default set of optional packages, you MAY delete these after installation
```

In addition to the preselected packages, certain additional software from the RHEL CDs MAY be installed without invalidating the evaluated configuration. The rules described in section §4.4 "Installation of additional software" of this guide MUST be followed to ensure that the security requirements are not violated.

3.2 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 §4.7.7 "Defining administrative accounts" of this guide for more information about creating administrative accounts.

Please refer to sections §4.7 "Managing user accounts" and §7.3 "Password policy" of this guide for more information on creating user accounts.

3.3 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.

3.4 Automated configuration of the system

The kickstart script installs the *cc-eal4-config-rhel56* RPM package and runs the *cc-config* script contained within that RPM package noninteractively.

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-rhel56* 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.

3.5 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 printers supported by CUPS, system using a parallel port or USB connection. In addition, network printers may also be attached to the system.

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.

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

3.6 Setting up Postfix

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 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.

3.7.1 GRUB boot loader configuration

This section applies to the x86_64 (Xeon EM64T or AMD Opteron) platform only.

The GRUB boot loader is highly configurable, and permits flexible modifications at boot time through a special-purpose command line interface. Please refer to the *grub(8)* man page or run `info grub` for more information.

- Use the `password` command in */boot/grub/menu.lst* to prevent unauthorized use of the boot loader interface. Using md5 encoded passwords is RECOMMENDED, run the command *grub-md5-crypt* to generate the encoded version of a password.
- Protect all menu entries other than the default RHEL boot with the `lock` option, so that the boot loader will prompt for a password when the user attempts to boot from other media (such as a floppy) or sets other non-default options for the boot process. To implement this, add a line containing just the keyword `lock` after the `title` entry in the */boot/grub/menu.lst* file.

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 §7.5 "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 security relevant extended attributes, such as ACLs. You **MAY** omit the `-acls` option 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.

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 the `su` 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.

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.

4.4 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 (EM64T)

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.

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 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 100. 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.
- 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/*.
- Programs or packages that add or require modifications to the SELinux policy **MUST NOT** be added.

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.

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.

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*, 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.

Note that use of the *usbfs* filesystem type is **NOT** permitted (and not needed) in the evaluated configuration.

A new file system 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 in ext3 format 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**.

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 Managing user accounts

4.7.1 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
```

4.7.2 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.

4.7.3 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 SHOULD 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.

4.7.4 Changing user properties

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

4.7.5 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_tally2.so` PAM module enforces automatic lockout after excessive failed authentication attempts. Use the program `pam_tally2` to view and reset the counter if necessary, as documented in the file `/usr/share/doc/pam-*/txts/README.pam_tally2`. Note that the `pam_tally2` 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
pam_tally2 --user jdoe

# set new password, and reset the counter
passwd jdoe
pam_tally2 --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.

4.7.6 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 - this call catches all processes
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.)
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
```

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.

4.7.7 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
```

4.8 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 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
```

4.9 Managing data objects

4.9.1 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.

4.9.2 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.

4.9.3 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).

4.10 Configuring object access rights

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

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

4.11 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 be 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.

4.12 Firewall configuration

You MAY enable, reconfigure, or disable the builtin network firewall as required. The network firewall and its security properties are beyond the scope of this guide and were not part of the evaluation.

Useful commands include:

```
# Disable firewall
lokkit --disabled -q
chkconfig --level=3 iptables off
```

```
# Enable firewall
lokkit --enabled \
    --port=22:tcp \
    --port=222:tcp \
    --port=80:tcp \
```



```
--port=25:tcp \
--port=500:udp \
--port=:esp \
--port=:ah \
-q
```

```
chkconfig --level=3 iptables on
```

Please refer to the *lokkit* help information (available by running `lokkit -help`) and the *iptables(8)* man page for more information.

4.13 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
```

4.14 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 normally employ */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 `/etc/profile` and `/etc/csh.login` contain the setting of this environment variable for Bourne shell and derivatives as well as Z-shells and its derivatives, respectively.

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 `libvirt` 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 `libvirt` 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:

- IBM System x Intel x86_64 bit
 - 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.

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 is not used as the Linux kernel interface allowing user space to utilize the hardware support is not accessible to normal users. The device file `/dev/kvm` provides 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` 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`.

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`. The configuration option that enables the automated labeling of the virtual machine process and the virtual machine's resources is set in `/etc/libvirt/qemu.conf` with the configuration value `dynamic_ownership` set to 1.

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.

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 hub. 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.

5.5 Disk space management

The evaluated configuration allows virtual machine administrators to configure various storage backends that are passed to the virtual machines as disk devices. All configurations offered by `libvirtd` are allowed in the evaluated configuration with the following exceptions:

- The virtual machine administrator MUST ensure in case SCSI disks including all disks accessed via the SCSI driver framework (such as USB disks, IDE disks accessed via `libata` indicated by their device file names start with `sd` instead of `hd`) shall be assigned as disk backends for virtual machines that a virtual machine has exclusive access to a SCSI disk.

For example, `/dev/sda1` shall be assigned to virtual machine A as disk backend. The virtual machine administrator now MUST NOT assign any other partition from `/dev/sda` to any other virtual machine. A virtual machine administrator MAY assign other SCSI disks to other virtual machines, such as `/dev/sdb1`.

- The virtual machine administrator MUST NOT assign LVM disk devices as disk backends for virtual machines.

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/ld.so.conf
/etc/localtime
/etc/login.defs
/etc/modprobe.conf
/etc/pam.d/*
/etc/passwd
/etc/rc.d/init.d/*
/etc/rc.d/init.d/auditd
/etc/securetty
/etc/security/opasswd
/etc/shadow
/etc/ssh/sshd_config
/etc/sysconfig/*
/etc/sysctl.conf
/var/log/lastlog
/var/log/tallylog
/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 file
boot.msg	Messages from system startup
lastlog	Last successful log in (see <code>lastlog(8)</code>)
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 subsystem, see <code>who(1)</code>

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.

6.3.1 Intended usage of the audit subsystem

The evaluated configuration specifies 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 system compliant with the Protection Profile of this evaluation **MUST** disable logins if the audit subsystem is not working. Please ensure that you are aware of the consequences if you enable auditing.

The evaluated configuration is 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. The evaluation requires 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.

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

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:

```
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.

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 dated stamped 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 §3.7 "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, can be found in the `/usr/share/doc/*/` directories. Use the `less(1)` pager to read it, for example:

```
/usr/share/doc/bash*/FAQ
```

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

A collection of How-To documents in HTML format can be found under `/usr/share/doc/howto/en/html` if the optional `howtoenh` package is installed.

Please see `/usr/share/doc/howto/en/html/Security-HOWTO` for security information. The HTML files can be read with the `elinks` browser.

The RHEL server documentation is also available in electronic form in the directories `/usr/share/doc/rhel*`.

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
```

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 scrollbar 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 scrollbar 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 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.

7.4.1 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.

7.4.2 General access control

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.

7.5 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 option *-acls* must be used with *tar*.

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

7.6 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
```


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.

RHEL5 Installation Guide: https://www.redhat.com/docs/manuals/enterprise/RHEL-5-manual/Installation_Guide-en-US/index.html

RHEL5 Deployment Guide: https://www.redhat.com/docs/manuals/enterprise/RHEL-5-manual/Deployment_Guide-en-US/index.html

David A. Wheeler, "Secure Programming for Linux and Unix HOWTO", /usr/share/doc/howto/en/html_single/Secure-Programs-HOWTO.html, <http://tldp.org/HOWTO/Secure-Programs-HOWTO/>

Kevin Fenzi, Dave Wreski, "Linux Security HOWTO", /usr/share/doc/howto/en/html_single/Security-HOWTO.html, <http://www.linuxsecurity.com/docs/LDP/Security-HOWTO/>

Libvirt configuration file guidance, <http://www.libvirt.org/format.html>

Richard Haines, "The SELinux Notebook - The Foundations", 2nd Edition, 2010, <http://www.freetechbooks.com/the-selinux-notebook-the-foundations-t785.html>

8.2 Literature

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

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