

# **Common Criteria EAL3+ Evaluated Configuration Guide for Red Hat Enterprise Linux on SGI Hardware**

Klaus Weidner <klaus@atsec.com>, Lori Gilbertson <loriann@sgi.com>

August 17, 2006; v2.7

atsec is a trademark of atsec GmbH

Altix and ProPack are trademarks or registered trademarks of Silicon Graphics Incorporated in the United States, other countries, or both.

ProLiant and Integrity are trademarks of Hewlett-Packard Company.

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

Intel, Itanium, Xeon, 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.

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 "EAL3 Evaluated Configuration Guide for Red Hat Enterprise Linux" Copyright (c) 2003, 2004 by atsec GmbH, and IBM Corporation or its wholly owned subsidiaries, and the "Common Criteria EAL3+ Evaluated Configuration Guide for Red Hat Enterprise Linux on HP Hardware" Copyright (c) 2004, 2005 by atsec inc., Hewlett-Packard Company or its wholly owned subsidiaries.

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

# Contents

<b>1</b>	<b>Introduction</b>	<b>5</b>
1.1	Purpose of this document . . . . .	5
1.2	How to use this document . . . . .	5
1.3	What is a CC compliant system? . . . . .	5
1.3.1	Hardware requirements . . . . .	6
1.3.2	Software requirements . . . . .	6
1.3.3	Environmental requirements . . . . .	6
1.3.4	Operational requirements . . . . .	6
1.4	Requirements for the system's environment . . . . .	7
1.5	Requirements for the system's users . . . . .	7
1.6	Overview of the system's security functions . . . . .	8
1.6.1	Identification and authentication . . . . .	8
1.6.2	Audit . . . . .	8
1.6.3	Discretionary access control . . . . .	8
1.6.4	Object reuse . . . . .	8
1.6.5	Security management and system protection . . . . .	9
1.6.6	Secure communication . . . . .	9
1.7	Overview of security relevant events . . . . .	9
<b>2</b>	<b>Installation</b>	<b>9</b>
2.1	Supported hardware . . . . .	9
2.2	Selection of install options and packages . . . . .	10
2.2.1	Prerequisites for installation . . . . .	10
2.2.2	Preparing for installation . . . . .	10
2.2.3	Customizing the installation . . . . .	12
2.2.4	Kickstart . . . . .	13
2.2.5	Pre-install configuration . . . . .	15
2.2.6	Partitioning . . . . .	15
2.2.7	Post-install configuration . . . . .	16
<b>3</b>	<b>Secure initial system configuration</b>	<b>17</b>
3.1	Creating additional user accounts for administrators . . . . .	17
3.2	Installing required updates . . . . .	17
3.3	Automated system configuration . . . . .	18
3.4	Configuring filesystem parameters . . . . .	18
3.4.1	Mount /proc nosuid . . . . .	19
3.4.2	Disable usbfs . . . . .	19
3.5	Add and remove packages . . . . .	20
3.6	Disable services . . . . .	20
3.7	Remove SUID/SGID root settings from binaries . . . . .	21
3.8	Update permissions for 'su' . . . . .	22
3.9	Configure root login . . . . .	22
3.10	Setting up SSH . . . . .	23
3.11	Setting up xinetd . . . . .	24
3.12	Setting up FTP . . . . .	24
3.13	Setting up Postfix . . . . .	25
3.14	Setting up CUPS . . . . .	25
3.15	Introduction to Pluggable Authentication Module (PAM) configuration . . . . .	26
3.16	Required Pluggable Authentication Module (PAM) configuration . . . . .	27
3.16.1	/etc/pam.d/system-auth . . . . .	28
3.16.2	/etc/pam.d/login . . . . .	29
3.16.3	/etc/pam.d/other . . . . .	29

3.16.4	/etc/pam.d/sshd . . . . .	30
3.16.5	/etc/pam.d/su . . . . .	30
3.16.6	/etc/pam.d/vsftpd . . . . .	31
3.17	Configuring default account properties . . . . .	31
3.18	Configuring the boot loader . . . . .	33
3.18.1	EFI boot loader configuration . . . . .	33
3.19	Reboot and initial network connection . . . . .	33
<b>4</b>	<b>System operation</b>	<b>33</b>
4.1	System startup, shutdown and crash recovery . . . . .	33
4.2	Backup and restore . . . . .	34
4.3	Gaining superuser access . . . . .	34
4.4	Editing configuration files . . . . .	35
4.5	Installation of additional software . . . . .	35
4.6	Scheduling processes using cron and at . . . . .	36
4.7	Mounting filesystems . . . . .	37
4.8	Managing user accounts . . . . .	38
4.9	Using serial terminals . . . . .	40
4.10	SYSV shared memory and IPC objects . . . . .	40
4.11	Configuring secure network connections with <i>stunnel</i> . . . . .	41
4.11.1	Introduction . . . . .	41
4.11.2	Creating an externally signed certificate . . . . .	42
4.11.3	Creating a self-signed certificate . . . . .	43
4.11.4	Activating the tunnel . . . . .	44
4.11.5	Using the tunnel . . . . .	46
4.11.6	Example 1: Secure SMTP delivery . . . . .	46
4.11.7	Example 2: Simple web server . . . . .	47
4.11.8	Example 1: system status view . . . . .	47
4.12	The Abstract Machine Testing Utility (AMTU) . . . . .	48
4.13	Setting the system time and date . . . . .	49
4.14	SELinux configuration . . . . .	49
<b>5</b>	<b>Monitoring, Logging &amp; Audit</b>	<b>49</b>
5.1	Reviewing the system configuration . . . . .	49
5.2	System logging and accounting . . . . .	51
5.3	Configuring the audit subsystem . . . . .	51
5.3.1	Intended usage of the audit subsystem . . . . .	51
5.3.2	Selecting the events to be audited . . . . .	52
5.3.3	Reading and searching the audit records . . . . .	52
5.3.4	Starting and stopping the audit subsystem . . . . .	53
5.3.5	Storage of audit records . . . . .	53
5.3.6	Reliability of audit data . . . . .	54
5.4	System configuration variables in <i>/etc/sysconfig</i> . . . . .	54
<b>6</b>	<b>Security guidelines for users</b>	<b>54</b>
6.1	Online Documentation . . . . .	54
6.2	Authentication . . . . .	55
6.3	Password policy . . . . .	56
6.4	Access control for files and directories . . . . .	57
6.5	Data import / export . . . . .	58
<b>7</b>	<b>Appendix</b>	<b>58</b>
7.1	Online Documentation . . . . .	58
7.2	Literature . . . . .	59

# 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 the terms "SHOULD" and "SHOULD NOT" are avoided in this document. 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 may 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 may be omitted from the command, they are only needed if pages with the same name exist in different sections,

## 1.3 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.

### 1.3.1 Hardware requirements

The hardware **MUST** be the one of the following SGI systems:

```
SGI Altix 400 series
SGI Altix 4000 series
```

consisting of a combination of the following blade types:

```
Compute/Memory blade
Memory-only blade
Base I/O Blade
PCI-X expansion blade
PCI-Express expansion blade
```

The hardware partition facility of Altix is not supported in the evaluated configuration and must not be used.

RASC blades are not supported in the evaluated configuration.

Examples of such servers are the ALTIX 450 and A4700.

Running the certified software on other similar hardware may 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.

### 1.3.2 Software requirements

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).

### 1.3.3 Environmental requirements

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.

For more information about these requirements, please refer to section §1.4 "Requirements for the system's environment" of this guide.

### 1.3.4 Operational requirements

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.

## 1.4 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 a directly Internet-connected server, or the case where services are to be provided to potentially hostile users.

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

Network access to the L1/L2 controllers is only allowed from within the physically secure environment.

You **MUST** ensure that all connections to peripheral devices and all network connections are protected against tampering, tapping and other modifications. Using the secured protocols SSHv2 or SSLv3 is considered sufficient protection for network connections. All other connections must remain completely within the physically secure server environment.

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.

If other systems are connected to the network they **MUST** be configured and managed by the same authority using an appropriate security policy not conflicting with the security policy of the target of evaluation. All links from this network to untrusted networks (such as the Internet) need to be protected by appropriate measures like carefully configured firewall systems that prevent attacks from the untrusted networks.

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).

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 the security 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.

## 1.5 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.

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.
- 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 §6 "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.

## 1.6 Overview of the system's security functions

This section summarizes the security functions that were covered by the evaluation. Please refer to the appropriate sections for information on configuring, using and managing these functions.

### 1.6.1 Identification and authentication

#### Pluggable Authentication Module (PAM)

Sections §3.15 "Introduction to Pluggable Authentication Module (PAM) configuration", §3.16 "Required Pluggable Authentication Module (PAM) configuration" of this guide; and the documentation in */usr/share/doc/pam\*/* and the *pam(8)* man page.

#### OpenSSH

Section §3.10 "Setting up SSH" of this guide; and the *sshd(8)*, *ssh(1)*, *sshd\_config(5)* man pages.

#### vsftpd

Section §3.12 "Setting up FTP" of this guide; and the *vsftpd(8)*, *vsftpd.conf(5)* man pages.

#### su

Sections §3.8 "Update permissions for 'su'", §4.3 "Gaining superuser access" of this guide; and the *su(8)* man page.

### 1.6.2 Audit

Section §5.3 "Configuring the audit subsystem" of this guide.

### 1.6.3 Discretionary access control

Sections §6.4 "Access control for files and directories" and §4.10 "SYSV shared memory and IPC objects" of this guide.

### 1.6.4 Object reuse

See the RHEL High Level Design document, the kernel automatically ensures that new objects (disk files, memory, IPC) do not contain any traces of previous contents.

### 1.6.5 Security management and system protection

Chapters §4 "System operation" and §5 "Monitoring, Logging & Audit".

### 1.6.6 Secure communication

Section §4.11 "Configuring secure network connections with *stunnel*" of this guide; and the *stunnel*(1) man page.

Section §3.10 "Setting up SSH" of this guide; and the *sshd*(8), *ssh*(1), and *sshd\_config*(5) man pages.

## 1.7 Overview of security relevant events

The audit subsystem is intended to be the central interface for collecting and viewing the record of security relevant events. The events being monitored by default in the evaluated configuration include:

- All authentication done through the PAM library, including the identity and location (where available) of the user and the success or failure result.
- Use of *su*(8) to change identity. All actions done as part of a *su* session are marked in the audit record with the original user's login user ID.
- Adding, changing, or deleting users or groups.
- Changes and change attempts to the contents of security critical files.
- Changes to the access permissions or ownership of any files or IPC objects.
- Binding network ports and accepting connections.

Please refer to section §5 "Monitoring, Logging & Audit" for more information.

## 2 Installation

The evaluation covers a fresh installation of RHEL AS, Version 4 Update 4, on one of the supported hardware platforms as defined in section §1.3.1 "Hardware requirements" of this guide.

The hardware partition facility of Altix is NOT supported in the evaluated configuration and MUST NOT be used.

### 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 and tape drives).
- All Ethernet and Token Ring network adapters supported by the operating system. Modems, ISDN and other WAN adapters are not part of the evaluated environment.
- Any printers supported by the operating system.

- Operator console consisting of a keyboard, video monitor, and optionally mouse. Additionally, you MAY directly attach supported serial terminals (see section §4.9 "Using serial terminals" of this guide), but *not* modems, ISDN cards, or other remote access terminals.

USB keyboards and mice MAY be attached, as some of the supported hardware platforms would otherwise not have supported console input devices. 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. You will need:

- The target system that will be installed, refer to section §1.3.1 "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 with a capacity of at least 32 MB, and 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://rhn.redhat.com/network/software/download\\_isos\\_full.pxt](https://rhn.redhat.com/network/software/download_isos_full.pxt) contains links to the platform-specific images.

You MUST use **Red Hat Enterprise Linux 4 Update 4 AS**. Make sure that you are using the appropriate version for your platform, refer to section §1.3.1 "Hardware requirements" of this guide for the list of supported hardware and the corresponding version needed.

You MUST verify that the MD5 checksums of the image files are correct. Run `md5sum *.iso` to view the checksums for the downloaded images, and compare them with those shown in this list:

Red Hat Enterprise Linux AS 4 Update 4 (Itanium)

```
4651b8e7882fab099986cca8235d9609 RHEL4-U4-ia64-AS-disc1.iso
9488bbbea7ed3bcd21988f33458a727 RHEL4-U4-ia64-AS-disc2.iso
2e2f6ecdb17a62872258bb6eb0d6d002 RHEL4-U4-ia64-AS-disc3.iso
ae0d6f1fe09fabebfbb8b2b5466f96b0 RHEL4-U4-ia64-AS-disc4.iso
a651052b67f638bc6bd90f2841fb3fe9 RHEL4-U4-ia64-AS-disc5.iso
```

You **MUST** download an additional package not included in Update 4 to set up the evaluated configuration. The package is available at the following location:

```
ftp://ftp.redhat.com/pub/redhat/linux/eal/EAL3_RHEL4/SGI/
```

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

You **MUST** select the appropriate RPM packages for your architecture. The 64bit architectures support execution of both 64bit and 32bit binaries, but this functionality is disabled for ia64/Itanium in the evaluated configuration.

#### ia64

This is a 64-bit-only platform. Use the **\*.ia64.rpm** or **\*.noarch.rpm** versions of packages. It can support 32bit i386 applications, but that functionality is disabled for the evaluated configuration.

The files needed are the *capp-eal3-config-sgi* RPM, and the unpacked kickstart file (contained within the *capp-eal3-config-sgi* RPM).

```
# ia64 (Itanium 2), AS
capp-eal3-config-sgi-*.rpm
```

You **MUST** have the Red Hat package signing key available to verify the integrity of the *capp-eal3-config-sgi* RPM package. It is available at the following location:

```
https://www.redhat.com/security/db42a60e.txt
```

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

```
rpm --import db42a60e.txt
rpm --checksig capp-eal3-config-sgi-*.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 *capp-eal3-config-sgi* RPM into a temporary directory:

```
mkdir capp-inst
cd capp-inst
rpm2cpio ../capp-eal3-config-sgi-*.rpm | cpio -i
```

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

```
# this guide, and supporting documentation
./usr/share/doc/capp-eal3-config-sgi-*/
    GPL.txt
    README-capp.txt
    RHEL-CAPP-EAL3-SGI-Configuration-Guide.*

# the kickstart configuration used to automate the installation
./usr/share/capp/kickstart/
    ks.cfg

# the evaluated configuration reconfiguration script
./usr/sbin/
    capp-eal3-config

# configuration files used for the evaluated configuration
./usr/share/capp/conf/
    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/capp/kickstart/` and the downloaded RPM packages, with all files at the top directory level (no subdirectories).
- Copy the kickstart files from `./usr/share/capp/kickstart/` and the downloaded RPM package onto a USB memory stick or USB external hard drive (with a capacity of at least 32 MB, and 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 and the kickstart file from `./usr/share/capp/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/New_York`

You MAY select a different time zone.

**firewall**

Default: `--enabled --port=22:tcp --port=80:tcp --port=21:tcp --port=25:tcp`

You MAY modify the firewall settings or disable the firewall.

**selinux**

Default: `--enforcing`

You MAY disable SELinux.

**## 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.

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, using the *ks.cfg* file provided inside the *capp-eal3-config-sgi* RPM file.

The installation process will prompt for all needed information, alternatively you MAY supply command line parameters to automate the installation. The pre-install configuration will be noninteractive if you supply both of the `ip` and `instdisk` parameters.

Use a combination of the following parameters as appropriate:

**method**

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

```
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=rhel4capp.example.com
```

(This parameter is specific to the CAPP kickstart install and not generally available)

**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
```

(This parameter is specific to the CAPP kickstart install and not generally available)

**console**

You MAY use a serial console to control the installation. Add the following parameters to activate the Altix serial console:

```
serial console=ttySG0
```

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.cfg method=cdrom serial console=ttySG0

# interactive network install, get IP address via DHCP
linux ks=http://example.com/rhel4/ks.cfg serial console=ttySG0

# noninteractive network install (enter on a single line)
linux ip=172.16.204.4 netmask=255.255.255.0 gateway=172.16.204.1
    dns=172.16.204.1
    ks=http://example.com/rhel4/ks.cfg
    method=cdrom
    hostname=rhel4.example.com
    instdisk=sda
    serial console=ttySG0
```

### 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 CAPP configuration kickstart ***

(Answer '!' at any prompt to get an interactive shell)

Installation source [cdrom] ?

Available destination disks:
sda (70001 MB)
sdb (34726 MB)

Install on which disk [sda] ?

Hostname (fully qualified) [rhel4capp.example.com] ?

Network interface [eth0] ?

IP address [] ? 192.168.1.4

Netmask [255.255.255.0] ?

Gateway [] ? 192.168.1.254

Nameserver list (comma separated) [] ? 192.168.1.3

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.

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

### 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 §5.3 "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.

The ia64 (Itanium) systems require a separate partition */boot/efi/* to contain the boot loader data and program files, which **MUST** be formatted using the **vfat** file system. On ia64 systems, there is no need for a separate */boot* partition in addition to the **REQUIRED** */boot/efi/* 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>	75 MB	# non-ia64, if needed by installer
<i>/boot/efi</i>	75 MB	# <b>REQUIRED</b> on ia64 as vfat partition
<i>/</i>	1200 MB	
<i>/tmp</i>	200 MB	
<i>/home</i>	100 MB	
<i>/var</i>	384 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**, except for */boot/efi/* which **MUST** be type *vfat*. Configuring a swap partition at least as large as the installed RAM is **RECOMMENDED**.

### 2.2.7 Post-install configuration

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

```
*** Common Criteria CAPP configuration kickstart ***

Please enter the password for the root account.
Changing password for user root.
New UNIX password:
Retype new UNIX password:

passwd(pam_unix)[5950]: password changed for root
passwd: all authentication tokens updated successfully.
Create an administrative user account.

Real name (First Last) [] ? John Smith

Userid [jsmith] ?
Changing password for user jsmith.
New UNIX password:
Retype new UNIX password:
passwd(pam_unix)[5958]: password changed for jsmith
passwd: all authentication tokens updated successfully.

Add more administrative users (y/n) [n] ?

Removing obsolete kernel devel packages...

Need to install the certification RPM and post-U2 RPM packages:

capp-eal3-config-sgi- [...]
```

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://partners.redhat.com/EAL3\_RHEL4/SGI/] ?

```
Preparing... ##### [100%]
  1:capp-eal3-config-sgi ##### [100%]
```

```
--- Wed Dec 14 11:06:15 EST 2005 script running: /usr/sbin/capp-eal3-config args -a
```

### 3 Secure initial system configuration

The automated kickstart post-install procedure reconfigures the system into the evaluated configuration noninteractively.

This section describes the steps done during the automated process for informational purposes. It is NOT recommended to attempt manually configuring the evaluated configuration.

You MAY use the *capp-eal3-config* after the initial installation script to re-run selected configuration steps automatically.

After software upgrades or installation of additional packages, these steps MUST be re-done or at least re-checked to ensure that the configuration remains secure.

#### 3.1 Creating additional user accounts for administrators

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

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

The kickstart script creates one or more administrative user accounts in the postinstall section.

Here is an example of creating an additional user account:

```
useradd -m -c "John Doe" -G wheel jdoe
passwd jdoe
chage -m 1 -M 60 -W 7 jdoe
```

Please refer to sections §4.8 "Managing user accounts" and §6.3 "Password policy" of this guide for more information on creating user accounts.

#### 3.2 Installing required updates

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

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

### 3.3 Automated system configuration

The kickstart script installs the `capp-eal3-config-sgi` RPM package and runs the `capp-eal3-config` script contained within that RPM package noninteractively.

You MAY run the `capp-eal3-config` script interactively after installation is complete to verify and reset configuration settings to appropriate values for the evaluated configuration.

The `capp-eal3-config-sgi` package contains EAL3 specific configuration files, and the script `capp-eal3-config` that sets up the evaluated configuration.

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

```
capp-eal3-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 `capp-eal3-config` script will reboot the system as the final step in the process, as described in the manual instructions in section §3.19 "Reboot and initial network connection" of this guide. Remember to remove any CD-ROM from the drive and/or configure the system to boot from hard disk only.

The remaining steps in this chapter were done automatically if the kickstart install has completed successfully, or if you have run the script at a later time. You MAY skip ahead to section §4 "System operation" of this guide.

### 3.4 Configuring filesystem parameters

You MUST add the mount option `acl` in the file `/etc/fstab` for all ext3 file systems. You MAY also add the option `user_xattr`. Multiple options are separated with commas (not "comma space"), for example `acl,user_xattr`.

Edit `/etc/fstab` and replace the `defaults` option specification (fourth column) with `acl` for all file systems with type `ext3` (third column). Then, run `mount MOUNTPOINT -o remount` for each of the mount points (second column).

On ia64 (Itanium) platforms only, the filesystem `/boot/efi/` MUST be mounted with the option `umask=077` to ensure that only administrators are able to access the data.

You MUST configure the `/proc` filesystem to be mounted with the `nosuid` option to prevent exploitation of the CVE-2006-3626 bug. (This entry is for reference, the boot process does not refer to the `/etc/fstab` file when mounting `/proc`. Refer to the next section for the corresponding modification to the boot script.)

Following are sample `/etc/fstab` entries to illustrate the format, but DO NOT simply copy this as the specific entries depend on the partitioning scheme:

#	device	mntpt	type	options	dump	pass
	<code>LABEL=/</code>	<code>/</code>	<code>ext3</code>	<code>acl,user_xattr</code>	<code>1</code>	<code>1</code>
	<code>LABEL=/var/log</code>	<code>/var/log</code>	<code>ext3</code>	<code>acl,user_xattr</code>	<code>1</code>	<code>2</code>
	<code>LABEL=/boot/efi</code>	<code>/boot/efi</code>	<code>vfat</code>	<code>umask=077</code>	<code>1</code>	<code>2</code>
	<code>none</code>	<code>/proc</code>	<code>proc</code>	<code>nosuid,nodiratime</code>	<code>0</code>	<code>0</code>

For more information, please refer to section §4.7 "Mounting filesystems" of this guide.

### 3.4.1 Mount /proc nosuid

You **MUST** configure the */proc* filesystem to be mounted with the *nosuid* option to prevent exploitation of the CVE-2006-3626 bug.

The */proc* filesystem is mounted in the */etc/rc.d/rc.sysinit* file, use the following command to verify the current content:

```
grep 'mount.*-t proc' /etc/rc.d/rc.sysinit
```

Here is the output of the *grep* command before modification:

```
mount -n -t proc /proc /proc
```

Either use a text editor to edit the files manually (add *-o nosuid,nodiratime* to the boot options), or use the following automated method:

```
perl -pi.bak -e 'if (/mount.*-t proc/) {
    s{.*}{mount -n -t proc -o nosuid,nodiratime /proc /proc/}
}' /etc/rc.d/rc.sysinit
```

After the modification, the file content **MUST** be as follows; verify by re-running the *grep* command:

```
mount -n -t proc -o nosuid,nodiratime /proc /proc/
```

You **MUST NOT** manually remove the *nosuid* mount option for the */proc* file system.

### 3.4.2 Disable usbfs

The *usbfs* file system is not permitted in the evaluated configuration and **MUST** be disabled. Note that the only permitted USB devices are keyboards and mice connected at boot, and these also work without *usbfs* for the supported hardware. Please refer to sections §2.1 "Supported hardware" and §4.7 "Mounting filesystems" of this guide for more information.

*usbfs* is activated in the */etc/rc.d/rc.sysinit* file, use the following command to verify the current content:

```
grep 'mount.*usb' /etc/rc.d/rc.sysinit
```

Here is the output of the *grep* command before modification:

```
[ -d /proc/bus/usb ] && mount -n -t usbfs /proc/bus/usb /proc/bus/usb
[ -f /proc/bus/usb/devices ] && mount -f -t usbfs usbfs /proc/bus/usb
```

Either use a text editor to edit the files manually (putting a *#* hash mark comment character at the start of each of the two lines containing the *usbfs* mount command), or use the following automated method:

```
perl -pi.bak -e 's/^/#/ if /mount.*usb/' /etc/rc.d/rc.sysinit
```

After the modification, the file content **MUST** be as follows; verify by re-running the *grep* command:

```
#[ -d /proc/bus/usb ] && mount -n -t usbfs /proc/bus/usb /proc/bus/usb
#[ -f /proc/bus/usb/devices ] && mount -f -t usbfs usbfs /proc/bus/usb
```

You **MUST NOT** manually mount the *usbfs* file system.

### 3.5 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.

The following optional packages MAY be deleted from the system:

```
audit-libs-devel
autoconf
automake
bison
cyrus-sasl-devel
expect
expect-devel
flex
gcc
gcc-c++
kernel-devel
libattr-devel
libselinux-devel
libuser-devel
make
openssl-devel
pam-devel
perl-Digest-HMAC
perl-Digest-SHA1
readline-devel
rpm-build
strace
tcl
texinfo
tk
zlib-devel
```

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 §4.5 "Installation of additional software" of this guide MUST be followed to ensure that the security requirements are not violated.

### 3.6 Disable services

Note: The system runlevel as specified in the 'initdefault' entry in */etc/inittab* MUST remain at the default setting of '3' for these steps to be valid.

The following services are REQUIRED for runlevel 3:

```
atd           # the 'at' daemon
auditd        # the audit daemon
crond         # vixie-cron
irqbalance    # configures SMP IRQ balancing
kudzu         # new device discovery
network       # network interface configuration
syslog        # system logging
```

The following services are OPTIONAL for runlevel 3:

```

cups          # print subsystem
gpm           # console mouse management
mdmonitor     # software raid monitoring
postfix       # SMTP MTA
rawdevices    # Raw partition management (eg. for Oracle)
sshd          # Secure Shell
vsftpd        # FTP server
xinetd        # Internet Services

```

You **MUST** ensure that all **REQUIRED** services are active. You **MAY** enable or disable services from the **OPTIONAL** list as suitable for your configuration. All other services **MUST** be deactivated.

Use *chkconfig* *SERVICENAME* *off* to disable a service, and *chkconfig* *SERVICENAME* *on* to enable it. The following command lists the active services:

```
chkconfig --list | grep "3:on" | sort
```

Make sure that the audit subsystem is activated. If *auditd* is not running, all logins are automatically disabled in the evaluated configuration as required by CAPP.

### 3.7 Remove SUID/SGID root settings from binaries

Use of the SUID bit on binaries (to run with root privileges, a.k.a. "setuid bit") **MUST** be limited to those shown in the following list:

```

/bin/ping
/bin/su
/usr/bin/at
/usr/bin/chage
/usr/bin/chfn
/usr/bin/chsh
/usr/bin/crontab
/usr/bin/gpasswd
/usr/bin/newgrp
/usr/bin/passwd

```

The other binaries that were installed with the SUID bit set **MUST** have this bit removed. Administrators can still run these binaries normally, but they are not available for ordinary users.

There are also a number of SGID files on the system that are needed:

```

/usr/sbin/postdrop    # group "maildrop"
/usr/sbin/postqueue   # group "maildrop"
/usr/sbin/utempter    # group "utmp"

```

Similarly, the SGID bit **MUST NOT** be used to give group "root" privileges to any binary.

Generate a list of all SUID/SGID programs on the system by running the following command:

```

find / -not -fstype ext3 -prune -o \
-type f \( -perm -4000 -o -perm -2000 \) \
-print

```

Then, for each file in this list that is not one of the permitted SUID or SGID programs, run the command *chmod -s FILE* to remove the SUID and SGID bits. When done, re-run the *find* command to verify that the list matches the permitted programs.

### 3.8 Update permissions for 'su'

The 'su' binary MUST be restricted to members of the trusted 'wheel' group. This will be enforced both with PAM configuration (configured later) and the binary's permissions.

```
chgrp wheel /bin/su
chmod 4710 /bin/su
```

You MUST have at least one user account other than 'root' configured to be a member of the 'wheel' group, otherwise system administration will ONLY be possible from the system console.

### 3.9 Configure root login

Login from the network with user ID 0 ('root') MUST NOT be permitted over the network. Administrators MUST use an ordinary user ID to log in, and then use the `/bin/su -` command to switch identities. For more information, refer to section §4.3 "Gaining superuser access" of this guide.

It is RECOMMENDED that you remind administrators of this by adding the following alias to the bash configuration file `/etc/profile` that disables the pathless 'su' command:

```
alias su="echo \"Always use '/bin/su -' (see Configuration Guide)\""
```

This alias can be disabled for the root user in `/root/.bashrc`:

```
unalias su
```

The restriction for direct root logins is enforced through two separate mechanisms. For network logins using ssh, the `PermitRootLogin no` entry in `/etc/ssh/sshd_config` MUST be set (see next section). Console and serial terminal logins use the `pam_securetty.so` PAM module in the `/etc/pam.d/login` file that verifies that the terminal character device used is listed in the file `/etc/securetty`.

The file `/etc/securetty` MUST NOT be changed from the secure default settings. The original contents are the following:

```
console
vc/1
vc/2
vc/3
vc/4
vc/5
vc/6
vc/7
vc/8
vc/9
vc/10
vc/11
tty1
tty2
tty3
tty4
tty5
tty6
tty7
```

```

tty8
tty9
tty10
tty11
# The following entries are not present on all systems
ttyS0
ttySG0

```

Note that some systems configure the serial console *ttyS0* as a secure terminal. If that is the case, you **MUST NOT** allow non-administrative users to use this serial terminal.

### 3.10 Setting up SSH

SSH protocol version 1 **MUST** be disabled. It has known security deficiencies.

The ssh client **MUST NOT** be set up SUID root (the SUID bit was removed in the post-install configuration). This prevents the use of some authentication methods normally supported by OpenSSH, but does not affect the evaluated configuration that uses password authentication exclusively.

The SSH Server **MUST** be configured to reject attempts to log in as root.

The permitted authentication mechanisms are per-user (nonempty) passwords and per-user DSS public key authentication. All other authentication methods **MUST** be disabled.

The setting `PAMAuthenticationViaKbdInt` **MUST** be disabled, since this would otherwise circumvent the disabled root logins over the network.

This results in the following option set for the SSH daemon that **MUST** be set in `/etc/ssh/sshd.config`:

```

# /etc/ssh/sshd_config
#
# CAPP/EAL3 configuration. Please read the Evaluated Configuration Guide
# before making changes.
#
# Cryptographic settings. Disallow the obsolete (and
# insecure) protocol version 1, and hardcode a strong
# cipher.
Protocol 2
Ciphers 3des-cbc

# Configure password-based login. This MUST use the PAM
# library exclusively, and turn off the builtin password
# authentication code.
UsePAM yes
ChallengeResponseAuthentication yes
PasswordAuthentication no
PermitRootLogin no
PermitEmptyPasswords no

# No other authentication methods allowed
IgnoreRhosts yes
RhostsRSAAuthentication no
HostbasedAuthentication no
PubkeyAuthentication no
RSAAuthentication no

```

```

KerberosAuthentication no
GSSAPIAuthentication no

# Other settings, MAY change "X11Forwarding" to "yes"
X11Forwarding no
Subsystem sftp /usr/lib/ssh/sftp-server

```

All other options **MUST NOT** be changed from the defaults or from those settings specified here. Specifically, you **MUST NOT** add other authentication methods (AFS, Kerberos, host-based) to those permitted here.

### 3.11 Setting up xinetd

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

The log method and the data that is to be logged are defined using the *defaults* entry in the */etc/xinetd.conf* file. The **RECOMMENDED** settings are:

```

defaults
{
    instances          = 60
    log_type           = FILE /var/log/xinetd.log
    log_on_success     = HOST PID EXIT DURATION
    log_on_failure     = HOST ATTEMPT
    cps                = 25 30
}

includedir /etc/xinetd.d

```

The *xinetd.conf(5)* man page contains more information on *xinetd* and configuration examples.

### 3.12 Setting up FTP

The evaluated configuration **OPTIONALLY** includes FTP services. Note that FTP does not provide support for encryption, so this is only **RECOMMENDED** for anonymous access to non-confidential files. If you do not specifically need FTP, it is **RECOMMENDED** that you disable the *vsftpd(8)* service.

Use the *chkconfig(8)* command to control the FTP service:

```

# Activate FTP service
chkconfig vsftpd on

# Disable FTP service
chkconfig vsftpd off

```

The *vsftpd* service uses several additional configuration files. In */etc/vsftpd/vsftpd.conf* the configuration of the ftp daemon is specified. In addition, the file */etc/vsftpd.ftpusers* is used for access control. Users listed in that file can **NOT** log in via FTP. This file initially contains all system IDs and the root user. It can be augmented with other IDs according to the local needs, but the *root* entry **MUST NOT** be removed. The *ftpusers* file is not checked by the ftp

daemon itself but by a PAM module. Please see section §3.16 "Required Pluggable Authentication Module (PAM) configuration" of this guide for details.

The setup of */etc/vsftpd/vsftpd.conf* depends on the local needs. Please refer to *vsftpd.conf(5)* for details.

The default configuration permits anonymous FTP. This setting is only suitable for distribution of public files for which no read access control is needed. The default configuration uses the following */etc/vsftpd/vsftpd.conf* settings:

```
anonymous_enable=YES
local_enable=NO
```

You MAY disable anonymous FTP with the following */etc/vsftpd/vsftpd.conf* setting:

```
anonymous_enable=NO
```

You MAY enable FTP authentication for local user accounts with the following */etc/vsftpd/vsftpd.conf* setting:

```
local_enable=YES
```

It is RECOMMENDED to use the more secure alternatives *sftp(1)* or *scp(1)* to copy files among users, and to use FTP only for legacy applications that do not support this alternative.

### 3.13 Setting up Postfix

You MUST disable the execution of user-specified programs when receiving mail via entries in the *\$HOME/.forward* files of individual users. Add the following line to the */etc/postfix/main.cf* file:

```
allow_mail_to_commands = alias
```

It is RECOMMENDED that you set up an alias for root in the */etc/aliases* file. Specify one or more user names of administrators to whom mail addressed to *root* will be forwarded, for example with the following entry in the */etc/aliases* file:

```
root: jdoe, jsmith
```

You must then rebuild the aliases database and reload the database using the following commands:

```
newaliases
postfix reload
```

Please see *postfix(1)*, *master(8)*, *aliases(5)*, *newaliases(1)*, and the documentation in */usr/share/doc/postfix\*/* for details.

### 3.14 Setting up CUPS

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

By default the *cupsd* daemon runs as user root, this is not acceptable for the evaluated configuration. You MUST reconfigure the service by putting the following settings in the */etc/cups/cupsd.conf* file:

```
User lp
Group sys
RunAsUser Yes
```

Verify that the printer daemon is able to access your printer devices with these 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)` and `cupsd(8)` man pages for more information.

### 3.15 Introduction to Pluggable Authentication Module (PAM) configuration

The PAM subsystem is responsible for maintaining passwords and other authentication data. Because this is a security-critical system, understanding how it works is very important. In addition to the `pam(8)` manual page, full documentation is available in `/usr/share/doc/pam-*/txts/` and includes *"The Linux-PAM System Administrator's Guide"* (`pam.txt`) as well as information for writing PAM applications and modules. Detailed information about modules is available in `/usr/share/doc/pam-*/txts/README.pam_*` as well as manual pages for individual modules, such as `pam_stack(8)`.

The PAM configuration is stored in the `/etc/pam.d/` directory. Note that the documentation refers to a file `/etc/pam.conf` that is not used by RHEL (PAM was compiled to ignore this file if the `/etc/pam.d/` directory exists).

Each service (application) that uses PAM for authentication uses a *service-name* to determine its configuration, stored in the `/etc/pam.d/SERVICE_NAME` file. The special *service-name* `OTHER` (case insensitive) is used for default settings if there are no specific settings.

The configuration file for the service contains one entry for each module, in the format:

```
module-type    control-flag    module-path    args
```

Comments MAY be used extending from `'#'` to the end of the line, and entries MAY be split over multiple lines using a backslash at the end of a line as a continuation character.

The *module-type* defines the type of action being done. This can be one of four types:

#### **auth**

Authenticates users (determines that they are who they claim to be). It can also assign credentials, for example additional group memberships beyond those specified through `/etc/passwd` and `/etc/groups`. This additional functionality MUST NOT be used.

#### **account**

Account management not related to authentication, it can also restrict access based on time of day, available system resources or the location of the user (network address or system console).

#### **session**

Manages resources associated with a service by running specified code at the start and end of the session. Typical usage includes logging and accounting, and initialization such as auto mounting a home directory.

#### **password**

Used for updating the password (or other authentication token), for example when using the `passwd(1)` utility to change it.

The *control-flag* specifies the action that will be taken based on the success or failure of an individual module. The modules are stacked (executed in sequence), and the *control-flags* determine which final result (success or failure) will be returned, thereby specifying the relative importance of the modules.

Stacked modules are executed in the order specified in the configuration file.

The *control-flag* can be specified as either a single keyword, or alternatively with a more elaborate syntax that allows greater control. RHEL uses only the single keyword syntax by default.

The following keywords control how a module affects the result of the authentication attempt:

#### **required**

If this module returns a failure code, the entire stack will return failure. The failure will be reported to the application or user only after all other modules in the stack have been run, to prevent leakage of information (for example, ask for a password even if the entered username is not valid).

#### **requisite**

Same as **required**, but return failure immediately not executing the other modules in the stack. Can be used to prevent a user from entering a password over an insecure connection.

#### **sufficient**

Return success immediately if no previous **required** modules in the stack have returned failure. Do not execute succeeding modules.

#### **optional**

The return code of this module is ignored, except if all other modules in the stack return an indeterminate result (PAM\_IGNORE).

The *module-path* specifies the filename of the module to be run (relative to the directory */lib/security/*, and the optional *args* are passed to the module - refer to the module's documentation for supported options.

### **3.16 Required Pluggable Authentication Module (PAM) configuration**

You **MUST** restrict authentication to services that are explicitly specified. The 'other' fallback **MUST** be disabled by specifying the *pam\_deny.so* module for each *module-type* in the 'other' configuration. This ensures that access decisions within the PAM system are handled only by the service specific PAM configuration.

Note that RHEL uses the *pam\_stack(8)* module to unify commonly used configuration options within single files, rather than having redundant information in multiple files. You **MUST** verify that the shared settings are applicable to services that use *pam\_stack*, and keep in mind that a change to the shared file will affect several services.

You **MUST** add the *pam\_wheel.so* module to the 'auth' *module\_type* configuration for the 'su' service to restrict use of *su(1)* to members of the 'wheel' group.

You **MUST** add the *pam\_tally.so* module to the auth and account *module\_type* configurations of *login*, *sshd* and *vsftpd*. This ensures that accounts are disabled after several failed login attempts. The *pam\_tally.so* module is used in the auth stack to increment a counter in the file */var/log/faillog*, and in the account stack to either deny login after too many failed attempts, or to reset the counter to zero after successful authentication. The evaluated configuration uses a lockout after five failed attempts, corresponding to the setting *deny=5*, you **MAY** decrease the number for stricter enforcement. Be aware that this can be used in denial-of-service attacks to lock out legitimate users. Please refer to section §4.8 "Managing user accounts" of this guide for more information.

You **MUST** use the *pam\_passwdqc.so* password quality checking module to ensure that users will not use easily-guessable passwords.

You **MUST** use the *pam\_loginuid.so* module for all authentication paths where human users are identified and authenticated, and add the *require\_auditd* option for all cases where the authentication method is accessible

to non-administrative users. This module sets the persistent login user ID and prevents login in case the audit system is inoperable for fail-secure operation.

The system supports many other PAM modules apart from the ones shown here. In general, you MAY add PAM modules that add additional restrictions. You MUST NOT weaken the restrictions through configuration changes of the modules shown here or via additional modules. Also, you MUST NOT add PAM modules that provide additional privileges to users (such as the *pam\_console.so* module).

You MUST NOT run the *authconfig(8)* tool to modify the authentication configuration.

Following are the pam configuration files:

### 3.16.1 /etc/pam.d/system-auth

This file contains common settings that are shared by multiple services using authentication. The *pam\_passwdqc.so* module is configured to enforce the minimum password length of 8 characters. Note that the *pam\_passwdqc.so* module is not part of a default installation, it was added previously as described in section §3.5 "Add and remove packages" of this guide.

The *pam\_tally* module MUST be used to block the user after 5 failed login attempts.

The *remember* option to *pam\_unix.so* prevents users from reusing old passwords. Hashes of old passwords are stored in the file */etc/security/opasswd* with the exception of passwords changed by the "root" user. Note that this file MUST exist, otherwise users cannot change passwords. Use the following commands to create it:

```
touch /etc/security/opasswd
chmod 600 /etc/security/opasswd
```

The file */etc/pam.d/system-auth* MUST be set up with the following content:

```
#%PAM-1.0
#
# pam.d/system-auth - PAM master configuration for CAPP compliance
# see the Evaluated Configuration Guide for more info
#

auth      required      pam_tally.so onerr=fail no_magic_root
auth      required      pam_env.so
auth      required      pam_unix.so likeauth nullok

account   required      pam_unix.so
account   required      pam_tally.so deny=5 reset no_magic_root

password  required      pam_passwdqc.so min=disabled,disabled,16,12,8 \
                        random=42
password  required      pam_unix.so nullok use_authtok md5 \
                        shadow remember=7

session   required      pam_limits.so
session   required      pam_unix.so
```

### 3.16.2 /etc/pam.d/login

This file configures the behavior of the *login* program. It allows root login only for terminals configured in */etc/securetty*. If the file */etc/nologin* is present, then only root can log in.

The *pam\_loginuid.so* module is by default configured without the *require\_auditd* option, which assumes that all terminals available for login are in physically secure locations and accessible only for authorized administrators. This permits administrators to log in on the console even if the audit subsystem is not available. If any serial terminals are attached and available for arbitrary users, you **MUST** add the *require\_auditd* option to ensure the CAPP-compliant fail-secure operating mode that disables login if audit is not working. Please refer to section §4.9 "Using serial terminals" of this guide for more information.

```
#%PAM-1.0
#
# pam.d/login - PAM login configuration for CAPP compliance
#               see the Evaluated Configuration Guide for more info
#
# If serial terminals are in use, pam_loginuid.so MUST use the 'require_auditd'
# option for CAPP-complaint fail-secure auditing. The default mode assumes that
# all terminals are in physically secure locations.
#

auth      required      pam_securetty.so
auth      required      pam_stack.so service=system-auth
auth      required      pam_nologin.so

account   required      pam_stack.so service=system-auth

password  required      pam_stack.so service=system-auth

# pam_selinux.so close should be the first session rule
session   required      pam_selinux.so close
session   required      pam_stack.so service=system-auth
session   optional      pam_console.so
# add 'require_auditd' option to pam_loginuid.so for fail-secure mode
session   required      pam_loginuid.so
# pam_selinux.so open should be the last session rule
session   required      pam_selinux.so multiple open
```

### 3.16.3 /etc/pam.d/other

This configuration applies for all PAM usage for which no explicit service is configured. It will log and block any attempts.

```
#%PAM-1.0
#
# pam.d/other - PAM other configuration for CAPP compliance
#               see the Evaluated Configuration Guide for more info
#

auth      required      pam_warn.so
auth      required      pam_deny.so
```

account	required	pam_warn.so
account	required	pam_deny.so
password	required	pam_warn.so
password	required	pam_deny.so
session	required	pam_warn.so
session	required	pam_deny.so

### 3.16.4 /etc/pam.d/sshd

This file configures the PAM usage for SSH. This is similar to the *login* configuration. The *securetty* entry is not applicable to network logins, and the *pam\_loginuid.so* module MUST be configured to prevent network login if the audit system is not available. Note that *pam\_loginuid.so* MUST run in the *account* stack, it does not work in the *account* or *auth* stacks due to the OpenSSH privilege separation mechanism.

```

#%PAM-1.0
#
# pam.d/sshd - pam.d/sshd configuration for CAPP compliance
#               see the Evaluated Configuration Guide for more info
#

auth          required    pam_stack.so service=system-auth
auth          required    pam_nologin.so

account       required    pam_stack.so service=system-auth

password      required    pam_stack.so service=system-auth

session       required    pam_stack.so service=system-auth
session       required    pam_loginuid.so require_auditd

```

### 3.16.5 /etc/pam.d/su

This file configures the behavior of the 'su' command. Only users in the trusted 'wheel' group can use it to become 'root', as configured with the *pam\_wheel* module.

```

#%PAM-1.0
#
# pam.d/su - PAM su configuration from CAPP compliance
#               see the Evaluated Configuration Guide for more info
#

auth          sufficient   pam_rootok.so
auth          required     pam_wheel.so use_uid
auth          required     pam_stack.so service=system-auth

account       required     pam_stack.so service=system-auth

password      required     pam_deny.so

```

```
# pam_selinux.so close must be first session rule
session    required    pam_selinux.so close
session    required    pam_stack.so service=system-auth
# pam_selinux.so open and pam_xauth must be last two session rules
session    required    pam_selinux.so open multiple
session    optional    pam_xauth.so
```

The *password* branch is disabled because forcing the root user to change the root password is not desired for this program,

### 3.16.6 /etc/pam.d/vsftpd

This file configures the authentication for the FTP daemon. With the listfile module, users listed in */etc/vsftpd.ftpusers* are denied FTP access to the system. Note that the setting is relevant only for authentication of incoming connections, and does not prevent local users from using the *ftp(1)* client to access other machines on the network.

```
##PAM-1.0
#
# pam.d/vsftpd - vsftpd configuration for CAPP compliance
#               see the Evaluated Configuration Guide for more info

auth          required    pam_listfile.so item=user sense=deny \
                        file=/etc/vsftpd.ftpusers onerr=succeed
auth          required    pam_stack.so service=system-auth
auth          required    pam_shells.so

account       required    pam_stack.so service=system-auth
account       required    pam_loginuid.so require_auditd

password      required    pam_deny.so

session       required    pam_stack.so service=system-auth
```

*pam\_deny.so* is used in the password stack because the FTP protocol has no provisions for changing passwords.

## 3.17 Configuring default account properties

The file */etc/login.defs* defines settings that will be used by user management tools such as *useradd(8)*. The file is not used during the authentication process itself.

The password aging settings defined in this file are used when creating users and when changing passwords, and stored in the user's */etc/shadow* entry. Note that only the */etc/shadow* entries are considered during authentication, so changes in */etc/login.defs* will not retroactively change the settings for existing users.

The *PASS\_MIN\_LEN* setting has no effect in the evaluated configuration, the relevant settings are instead configured using the *min=* parameter to *pam\_passwdqc.so* in the */etc/pam.d/system-auth* file.

```
### /etc/login.defs
# Global user account settings for the CAPP/EAL3 evaluated configuration.
#
# Mailbox settings:
```

```

#   Directory where mailboxes reside, _or_ name of file, relative to the
#   home directory. If you _do_ define both, MAIL_DIR takes precedence.
#   QMAIL_DIR is for Qmail
#
#   The setting is used only when creating or deleting users, and has
#   no effect on the mail delivery system. MAY be changed as required.
#
#QMAIL_DIR      Maildir
#MAIL_FILE      .mail
MAIL_DIR        /var/spool/mail
#
# Password aging controls:
#
#   PASS_MAX_DAYS      Maximum number of days a password may be used.
#   PASS_MIN_DAYS      Minimum number of days allowed between password changes
#   PASS_MIN_LEN       Minimum acceptable password length.
#   PASS_WARN_AGE      Number of days warning given before a password expires.
#
PASS_MAX_DAYS   60    # MAY be changed, must be <= 60
PASS_MIN_DAYS   1     # MAY be changed, 0 < PASS_MIN_DAYS < PASS_MAX_DAYS
PASS_WARN_AGE   7     # MAY be changed
PASS_MIN_LEN     5    # no effect in the evaluated configuration
#
# Min/max values for automatic uid selection in useradd
#
# MAY be changed, 100 < UID_MIN < UID_MAX < 65535
#
UID_MIN          500
UID_MAX          60000
#
# Min/max values for automatic gid selection in groupadd
#
# MAY be changed, 100 < GID_MIN < GID_MAX < 65535
#
GID_MIN          500
GID_MAX          60000
#
# If defined, this command is run when removing a user.
# It should remove any at/cron/print jobs etc. owned by
# the user to be removed (passed as the first argument).
#
# MAY be activated as described in the "Managing user accounts"
# section of the ECG.
#
#USERDEL_CMD     /usr/sbin/userdel_local
#
# If useradd should create home directories for users by default
# On RH systems, we do. This option is ORed with the -m flag on
# useradd command line.
#
# MAY be changed.
#
CREATE_HOME      yes

```

### 3.18 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.

#### 3.18.1 EFI boot loader configuration

This section applies to the ia64 (Itanium) platform only.

On this platform, the filesystem `/boot/efi/` is of type `vfat` with no built-in access control support. It **MUST** be mounted with the option `umask=077` to ensure that only administrators are able to access the data.

For more information on setting this mount option, please refer to section §3.4 "Configuring filesystem parameters" of this guide.

Installing kernel RPM packages will automatically configure the necessary data in `/boot/efi/` so that the builtin EFI boot loader will start the new kernel.

The file `/boot/efi/efi/redhat/elilo.conf` configures the boot options, including which kernel is booted and what kernel command line options are set. Please refer to the files `/usr/share/doc/elilo-*/*` for more information.

### 3.19 Reboot and initial network connection

*– This concludes the sections covered by the automated configuration script –*

After all the changes described in this chapter have been done, you **MUST** reboot the system to ensure that all unwanted tasks are stopped, and that the running kernel, modules and applications all correspond to the evaluated configuration.

Please make sure that the boot loader is configured correctly for your platform.

Remember to remove any CD-ROM from the drive and/or configure the system to boot from hard disk only.

The system will then match the evaluated configuration. The server **MAY** then be connected to a secure network as described above.

## 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)` 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, on systems using the *grub* boot loader you can use the following 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 *star*(1) archiver is RECOMMENDED for backups of complete directory contents, please refer to section §6.5 "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/
/var/spool/at/
```

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 §5.2 "System logging and accounting" and §5.3 "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* and *stunnel* servers, 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 superuser access

System administration tasks require superuser privileges. Since directly logging on over the network as user 'root' is disabled, 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 Editing configuration files

It is **RECOMMENDED** to use the *nano* editor to edit system configuration files. It is documented in the *nano(8)* man page.

It is **RECOMMENDED** to set the EDITOR environment variable in the */etc/profile* file to ensure that this editor is used by default by tools such as *crontab(1)*:

```
if [ `id -u` -eq 0 ]; then
    EDITOR=nano
    export EDITOR
fi
```

Other editors can read and execute commands specified in run control files or in the edited files themselves or have other complex dependencies on the execution environment. That is a security risk and inappropriate for an editor used for system administration tasks. The *nano* editor was tested and examined as part of the evaluation to ensure that it works as expected, this was not done for any of the other editors shipped with the system.

#### 4.5 Installation of additional software

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

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 500. This restriction is necessary to avoid conflict with the system user and group IDs such as the "bin" 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) or `at`

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.

## 4.6 Scheduling processes using cron and at

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.

The `at(1)` and `batch(1)` programs execute a command line at a specific single point of time. The same rules apply as for jobs scheduled via `cron(8)`. Use `atq(1)` and `atrm(1)` to manage the scheduled jobs.

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

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

```
/etc/at.allow
/etc/at.deny
/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.

The *root* user is always permitted to use the `crontab(1)` program on RHEL systems, even if listed in the `/etc/cron.deny` file.

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` and `at`.

It is **RECOMMENDED** to restrict the use of `cron` and `at` 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<500) print $1}' /etc/passwd >/etc/at.deny
chmod 600 /etc/at.deny
cp /etc/at.deny /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 */etc/at.allow* and/or */etc/cron.allow* files to explicitly list users who are permitted to use these services. If you do create these files, they MUST be owned by the user 'root' and have file permissions 0600 (no access for group or others).

## 4.7 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, please refer to section §3.4.2 "Disable usbfs" of this guide for more information

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 DVDs or CD-ROMs.

CD/DVD devices MUST be accessed using the *iso9660* filesystem type. Using the *udf* filesystem or 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 may 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 removeable 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.

## 4.8 Managing user accounts

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 **MUST** 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 §6.3 "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.

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
```

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.

You MAY use the *usermod*(8) command to change a user's properties. 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 if necessary
usermod -G $(groups jdoe | sed 's/.*: //; s/ /,/g'),wheel jdoe
```

Users MAY be locked out (disabled) using *passwd -l USER*, and re-enabled using *passwd -u USER*.

The *pam\_tally.so* PAM module enforces automatic lockout after excessive failed authentication attempts, as described in section §3.16 "Required Pluggable Authentication Module (PAM) configuration" of this guide. Use the program *pam\_tally* to view and reset the counter if necessary, as documented in the file */usr/share/doc/pam-\*/txts/README.pam\_tally*. Note that the *pam\_tally* 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_tally --user jdoe

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

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

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

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

# Kill all user processes, repeat if needed (or reboot)
kill -9 `ps -la --User $U|awk '{print $4}'`

# 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 and at jobs
crontab -u $U -r
find /var/spool/at -user $U -exec rm {} \;

# 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.

Group passwords are NOT supported in the evaluated configuration, and have been disabled by removing the SUID bit from the *newgrp*(8) program. You MUST NOT re-enable this feature and MUST NOT use *passwd*(1) with the *-g* switch or the *gpasswd*(1) command to set group passwords.

## 4.9 Using serial terminals

You MAY attach serial terminals to the system. They 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 CAPP-compliant fail-safe audit mode, you MUST ensure that the file */etc/pam.d/login* uses the option *require\_auditd* for the *pam\_loginuid.so* module in the *session* stack. Please refer to section §3.16.2 "etc/pam.d/login" of this guide for more information about the needed PAM configuration.

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 a 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
```

## 4.10 SYSV shared memory and IPC objects

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

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

## 4.11 Configuring secure network connections with *stunnel*

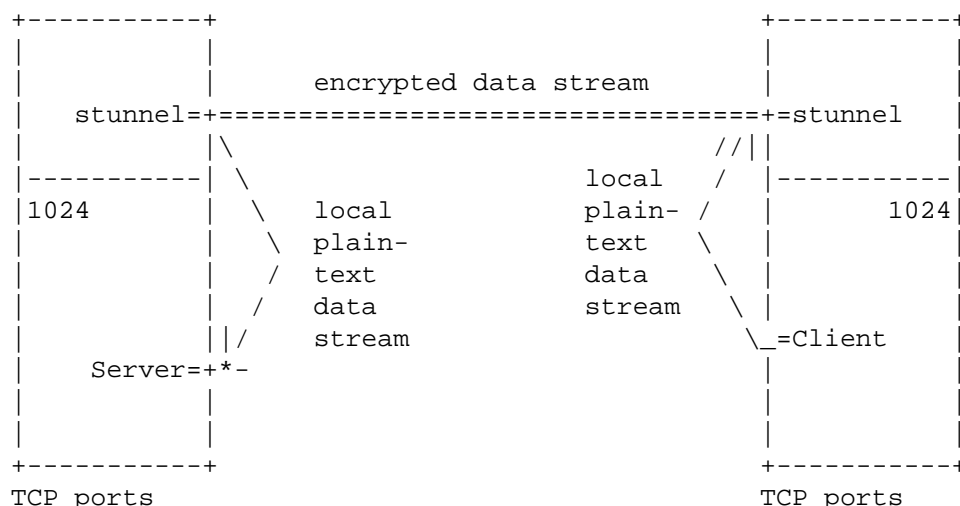
### 4.11.1 Introduction

The *stunnel* program is a flexible and secure solution for setting up encrypted network connections, enabling the use of strong encryption even for applications that are not able to use encryption natively. *stunnel* uses the OpenSSL library for its encryption functions, and the corresponding *openssl(1)* command line tool for key management.

*Stunnel* has three main operating modes:

- Accept incoming SSL-encrypted TCP connections, and run a specific program to handle the request.  
This is similar to how *xinetd* launches programs, and any program compatible with *xinetd* can also be used for this purpose. It must read and write the communication data on the *stdin* and *stdout* file descriptors and stay in the foreground. *stunnel* also supports switching user and group IDs before launching the program.
- Open a SSL connection to a remote SSL-capable TCP server, and copy data to and from *stdin* and *stdout*.
- Bind a TCP port to accept incoming unencrypted connections, and forward data using SSL to a prespecified remote server.

The following diagram shows a sample usage scenario:



In this scenario, neither the client nor the server have administrator privileges, they are running as normal user processes. Also, the client and server do not support encryption directly.

*stunnel* makes a secure communication channel available for the client and server. On the client, *stunnel* is accepting connections on TCP port 82. The client connects to this port on the local machine using normal unencrypted TCP, *stunnel* accepts the connection, and opens a new TCP connection to the *stunnel* server running on the remote machine. The *stunnel* instances use cryptographic certificates to ensure that the data stream has not been intercepted or tampered with, and then the remote *stunnel* opens a third TCP connection to the server, which is again a local unencrypted connection.

Any data sent by either the client or server is accepted by the corresponding *stunnel* instance, encrypted, sent to the other *stunnel*, decrypted and finally forwarded to the receiving program. This way, no modifications are required to the client and server.

To set up a secure connection compliant with the evaluated configuration, you **MUST** start the *stunnel* server(s) with administrator rights, and you **MUST** use a TCP port in the administrator-reserved range 1-1023 to accept incoming connections. A corresponding client which connects to the server **MAY** be started by any user, not just administrators.

*stunnel* MAY also be used by non-administrative users to receive encrypted connections on ports in the range 1024-65536. This is permitted, but it is outside of the scope of the evaluated configuration and not considered to be a trusted connection.

Any network servers and clients other than the trusted programs described in this guide (*stunnel*, *sshd*, *vsftpd*, *postfix* and *cupsd*) MUST be run using non-administrator normal user identities. Programs run from *stunnel* MUST be switched to a non-root user ID by using the *setuid* and *setgid* parameters in the */etc/stunnel/\*.conf* configuration files.

It is RECOMMENDED configuring any such servers to accept connections only from machine-local clients, either by binding only the *localhost* IP address 127.0.0.1, or by software filtering inside the application. This ensures that the only encrypted connections are possible over the network. Details on how to do this depend on the software being used and are beyond the scope of this guide.

Please refer to the *stunnel*(8) and *openssl*(1) man pages for more information.

#### 4.11.2 Creating an externally signed certificate

It is strongly RECOMMENDED that you have your server's certificate signed by an established Certificate Authority (CA), which acts as a trusted third party to vouch for the certificate's authenticity for clients. Please refer to the *openssl*(1) and *req*(1) man pages for instructions on how to generate and use a certificate signing request.

Create the server's private key and a certificate signing request (CSR) with the following commands:

```
touch /etc/stunnel/stunnel.pem

chmod 400 /etc/stunnel/stunnel.pem

openssl req -newkey rsa:1024 -nodes \
    -keyout /etc/stunnel/stunnel.pem -out /etc/stunnel/stunnel.csr
```

You will be prompted for the information that will be contained in the certificate. Most important is the "Common Name", because the connecting clients will check if the hostname in the certificate matches the server they were trying to connect to. If they do not match, the connection will be refused, to prevent a 'man-in-the-middle' attack.

Here is a sample interaction:

```
Generating a 1024 bit RSA private key
.....+++++
.....+++++
writing new private key to '/etc/stunnel/stunnel.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [PL]:US
State or Province Name (full name) [Some-State]:TX
Locality Name (eg, city) []:Austin
Organization Name (eg, company) [Stunnel Developers Ltd]:Example Inc.
Organizational Unit Name (eg, section) []:
Common Name (FQDN of your server) []:www.example.com
Common Name (default) []:localhost
```

The file `/etc/stunnel/stunnel.pem` will contain both the certificate (public key) and also the secret key needed by the server. The secret key will be used by non-interactive server processes, and cannot be protected with a passphrase. You **MUST** protect the secret key from being read by unauthorized users, to ensure that you are protected against someone impersonating your server.

Next, send the generated CSR file `/etc/stunnel/stunnel.csr` (*not* the private key) to the CA along with whatever authenticating information they require to verify your identity and your server's identity. The CA will then generate a signed certificate from the CSR, using a process analogous to `openssl req -x509 -in stunnel.csr -key CA-key.pem -out stunnel.cert`.

When you receive the signed certificate back from the CA, append it to the file `/etc/stunnel/stunnel.pem` containing the private key using the following command:

```
echo >> /etc/stunnel/stunnel.pem
cat stunnel.cert >> /etc/stunnel/stunnel.pem
```

Make sure that the resulting file contains no extra whitespace or other text in addition to the key and certificate, with one blank line separating the private key and certificate:

```
-----BEGIN RSA PRIVATE KEY-----
MIICXQIBAAKBgQCzF3ezbZFLjgv1YHNXnBnI8jmeQ5MmkvdNw9XkLnA2ONKQmvPQ
[ ... ]
4tjzwTFxPKYvAW3DnXxRAkAvaf1mbc+GTMoAiepXPVfqSpW2Qy5r/wa04d9phD5T
oUNbDU+ezu0Pana7mmmvG3Mi+BuqwlQ/iU+G/qrG6VGj
-----END RSA PRIVATE KEY-----

-----BEGIN CERTIFICATE-----
MIIC1jCCAj+gAwIBAgIBADANBgkqhkiG9w0BAQQFAFBXMQswCQYDVQQGEwJQTDET
[ ... ]
bIbYKL6Q1kE/vhGmRXcXQrZzkfu8sgJv7JsDpoTpAdUnmvssUY0bchqFo4Hhzkvs
U/whL2/8RFv5jw==
-----END CERTIFICATE-----
```

You **MAY** distribute the original signed certificate (`stunnel.cert` in this example) to clients, it does not contain any confidential information. *Never* distribute the file containing the private key, that is for use by the `stunnel` server only.

When using externally signed certificates, you **MUST** use the option `CPath` in `stunnel` client configuration files along with the setting `verify=2` or `verify=3` to enable the clients to verify the certificate.

#### 4.11.3 Creating a self-signed certificate

Alternatively, you **MAY** use a self-signed certificate instead of one signed by an external CA. This saves some time and effort when first setting up the server, but each connecting client **MUST** manually verify the certificate's validity. Experience shows that most users will not do the required checking and simply click "OK" for whatever warning dialogs that are shown, resulting in significantly reduced security. Self-signed certificates can be appropriate for controlled environments with a small number of users, but are not recommended for general production use.

Create a self-signed host certificate with the following commands:

```
# create secret key and self-signed certificate
```

```

openssl req -newkey rsa:1024 -nodes \
  -keyout /etc/stunnel/stunnel.pem \
  -new -x509 -sha1 -days 365 \
  -out /etc/stunnel/stunnel.cert

# set appropriate file permissions
chmod 400 /etc/stunnel/*.pem
chmod 444 /etc/stunnel/*.cert

# append copy of certificate to private key
echo >> /etc/stunnel/stunnel.pem
cat /etc/stunnel/stunnel.cert >> /etc/stunnel/stunnel.pem

```

The secret key contained in the */etc/stunnel/stunnel.pem* file **MUST** be kept secret.

You **MAY** distribute the public certificate contained in the */etc/stunnel/stunnel.cert* file to clients. Make sure you do not accidentally distribute the secret key instead.

The client has no independent way to verify the validity of a self-signed certificate, each client **MUST** manually verify and confirm the validity of the certificate.

One method is to give a copy of the self-signed certificate to the client (using a secure transport mechanism, not e-mail), and import it into the client directly. The *stunnel* client uses the *CAfile* option for this purpose.

Alternatively, many client programs (not *stunnel*) can interactively import the certificate when connecting to the server. The client will display information about the server's certificate including an MD5 key fingerprint. You **MUST** compare this fingerprint with the original fingerprint of the server's certificate.

Run the following command on the server to display the original certificate's fingerprint:

```
openssl x509 -fingerprint -in /etc/stunnel/stunnel.cert
```

Most clients will store the certificate for future reference, and will not need to do this verification step on further invocations.

#### 4.11.4 Activating the tunnel

In the evaluated configuration, you **MUST** use one of the following cipher suite suites as defined in the SSL v3 protocol:

# Cipher	Proto	Key	Authen-	Encryption	Message
#		exchg	tication		auth code
#					
RC4-SHA	SSLv3	Kx=RSA	Au=RSA	Enc=RC4(128)	Mac=SHA
DES-CBC3-SHA	SSLv3	Kx=RSA	Au=RSA	Enc=3DES(168)	Mac=SHA1
AES128-SHA	SSLv3	Kx=RSA	Au=RSA	Enc=AES(128)	Mac=SHA1
AES256-SHA	SSLv3	Kx=RSA	Au=RSA	Enc=AES(256)	Mac=SHA1

You **MUST** specify the cipher list in all *stunnel* client and server configuration files:

```

ciphers = RC4-SHA:DES-CBC3-SHA:AES128-SHA:AES256-SHA
options = NO_TLSv1
options = NO_SSLv2

```

For a service or tunnel that will only be used temporarily, simply launch the `stunnel` program from the command line and specify an appropriate configuration file. The tunnel will be available for multiple clients, but will not be started automatically after a reboot. To shut down the tunnel, search for the command line in the `ps ax` process listing, and use the `kill(1)` command with the PID shown for the *stunnel* process.

The RECOMMENDED method is to use two separate configuration files, one for server definitions (incoming connections use SSL), and one for client definitions (outgoing connections use SSL). More complex configurations will require additional configuration files containing individual service-specific settings. You MUST use the REQUIRED settings in all *stunnel* configuration files.

Use the following content for the file `/etc/stunnel/stunnel-server.conf`:

```
### /etc/stunnel/stunnel-server.conf
#
# The following settings are REQUIRED for CAPP compliance when used
# as a server, see ECG. File names MAY be changed as needed.
cert = /etc/stunnel/stunnel.pem
ciphers = RC4-SHA:DES-CBC3-SHA:AES128-SHA:AES256-SHA
options = NO_TLSv1
options = NO_SSLv2
#
# User and group ID MUST NOT be "root", but MAY be changed as needed.
setuid = nobody
setgid = nobody
#
# The following settings are RECOMMENDED
debug = 6
output = /var/log/stunnel-server.log
pid =
foreground = yes
#
# Individual service definitions follow
```

Use the following content for the file `/etc/stunnel/stunnel-client.conf`:

```
### /etc/stunnel/stunnel-client.conf
#
# The following settings are REQUIRED for CAPP compliance when used
# as a client, see ECG. File names MAY be changed as needed. You
# MAY use CAspath instead of CAfile for externally signed certificates.
CAfile = /etc/stunnel/stunnel.cert
ciphers = RC4-SHA:DES-CBC3-SHA:AES128-SHA:AES256-SHA
options = NO_TLSv1
options = NO_SSLv2
client = yes
verify = 2
#
# User and group ID MUST NOT be "root", but MAY be changed as needed.
setuid = nobody
setgid = nobody
#
# The following settings are RECOMMENDED
debug = 6
output = /var/log/stunnel-client.log
```

```
pid =
foreground = yes
#
# Individual service definitions follow
```

The RECOMMENDED launch method for *stunnel*(8) is via the *init*(8) process. This requires adding new entries to */etc/inittab*, the tunnels will be re-launched automatically whenever they are terminated, as well as after a reboot. The following are the RECOMMENDED */etc/inittab* entries:

```
ts:3:respawn:/usr/sbin/stunnel /etc/stunnel/stunnel-server.conf
tc:3:respawn:/usr/sbin/stunnel /etc/stunnel/stunnel-client.conf
```

Make sure you use the option `foreground = yes` in the configuration file when running from *init* (otherwise *init* will misinterpret the backgrounded server as having died and will try to restart it immediately, causing a loop), and use the `output` option to redirect the output to a log file.

#### 4.11.5 Using the tunnel

If the client program supports SSL encryption, it will be able to communicate with the *stunnel* service directly. You MUST verify and accept the server's certificate if the client cannot recognize it as valid according to its known certification authorities.

If the client program does not support SSL directly, you can use *stunnel* as a client, or indirectly by setting up a proxy that allows the client to connect to an unencrypted local TCP port.

**WARNING:** The *stunnel* client does *not* verify the server's certificate by default. You MUST specify either `verify = 2` or `verify = 3` in the client configuration file to switch on certificate verification.

You MAY also activate client certificate verification in the server's configuration file, so that the server can verify the client's identity as well.

You MUST specify the ciphers supported in the evaluated configuration in the configuration file as described in the previous section.

#### 4.11.6 Example 1: Secure SMTP delivery

Normal SMTP e-mail delivery is not encrypted, but most mail clients support the enhanced SMTPS protocol that uses SSL encryption. The protocol itself is unchanged other than being encrypted.

*stunnel* can easily be used as a proxy to receive SMTPS connections on the standard port expected by clients (465/tcp), and then forward the data to the mail server listening on the SMTP port (25/tcp). The mail server configuration does not need to be modified to support encryption of incoming mail.

To implement SSL support for incoming mail, add the following service definition to the */etc/stunnel/stunnel-server.conf* configuration:

```
[inbound_mail]
accept = 465
connect = 127.0.0.1:25
```

#### 4.11.7 Example 2: Simple web server

The following shell script acts as a simple web server, reading requests from standard input and writing HTTP/HTML to standard output:

```
cat > /usr/local/sbin/webserver_test <<__EOF__
#!/bin/sh
# Simple web server, can be run via stunnel or xinetd
#
# read and discard client data
dd bs=65536 count=1 >/dev/null 2>&1
#
# Send HTTP header
echo -e "HTTP/1.0 200\r"
echo -e "Content-type: text/html\r"
echo -e "\r"
#
# Send HTML output
echo "<html>"
echo "<h1>Test Page</h1>"
date
echo "<h2>Memory usage</h2>"
echo "<pre>"
free
echo "</pre>"
echo "</html>"
__EOF__

chmod +x /usr/local/sbin/webserver_test
```

Add the following entry to the */etc/stunnel/stunnel-server.conf* configuration to make this service available using the encrypted HTTPS protocol:

```
[webserver_test]
accept = 443
exec = /usr/local/sbin/webserver_test
TIMEOUTclose = 0
```

Then, use a SSL-capable web browser to connect to port 443:

```
elinks https://localhost/
```

#### 4.11.8 Example 1: system status view

This example shows how to combine *stunnel* client and server definitions to implement an encrypted tunnel for applications that do not themselves support encryption.

First, on the server machine, set up a *stunnel* server definition that accepts SSL connections on TCP port 444, and reports memory usage statistics for the server to connecting clients. Add the following service definition to the */etc/stunnel/stunnel-server.conf* configuration:

```
[free]
accept = 444
exec = /usr/bin/free
execargs = free
```

Then, on the client machine, add the following entry to the `/etc/stunnel/stunnel-client.conf` configuration, using the server's IP address instead of "127.0.0.1":

```
[free]
accept  = 81
connect = 127.0.0.1:444
```

On the client machine, connect to the local *stunnel* proxy by running the following command as a normal user:

```
telnet localhost 81
```

This will open an unencrypted TCP connection to the client's local port 81, then *stunnel* builds an encrypted tunnel to the server's port 444 and transfers the decrypted data (in this case, the "free" output) back to the client. All unencrypted connections are machine local, and the data transferred over the network is encrypted.

## 4.12 The Abstract Machine Testing Utility (AMTU)

The security of the operating system depends on correctly functioning hardware. For example, the memory subsystem uses hardware support to ensure that the memory spaces used by different processes are protected from each other.

The Abstract Machine Testing Utility (AMTU) is distributed as an RPM, and was installed previously as described in section §3.5 "Add and remove packages" of this guide.

To run all supported tests, simply execute the `amtu` program:

```
amtu
```

A successful run is indicated by the following output:

```
Executing Memory Test...
Memory Test SUCCESS!
Executing Memory Separation Test...
Memory Separation Test SUCCESS!
Executing Network I/O Tests...
Network I/O Controller Test SUCCESS!
Executing I/O Controller - Disk Test...
I/O Controller - Disk Test SUCCESS!
Executing Supervisor Mode Instructions Test...
Privileged Instruction Test SUCCESS!
```

The program will return a nonzero exit code on failure, which MAY be used to automatically detect failures of the tested systems and take appropriate action.

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

### 4.13 Setting the system time and date

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

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

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

```
date 050113002004
```

The `hwclock(8)` can query and modify the hardware clock on supported platforms. The typical use is to copy the current value of the software clock to the hardware clock. Note that the hardware clock **MAY** be running in either local time or universal time, as indicated by the `UTC` setting in the `/etc/sysconfig/clock` file. The following command sets the hardware clock to the current time using UTC:

```
hwclock -u -w
```

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

### 4.14 SELinux configuration

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

The CAPP/EAL3+ evaluation did not test SELinux features and does not provide any assurance related to SELinux functionality.

The `/etc/selinux/config` file has the following content by default:

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

You **MAY** disable SELinux by using one of the settings `SELINUX=disabled` or `SELINUX=permissive` instead, or configure a different policy, but any additional restrictions added by SELinux are beyond the scope of the CAPP/EAL3+ security target. (Note that reconfiguring the SELinux policy is likely to affect your support contract status. This is also beyond the scope of this document.)

## 5 Monitoring, Logging & Audit

### 5.1 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 may 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/at.allow
/etc/at.deny
/etc/audit.rules
/etc/auditd.conf
/etc/cron.allow
/etc/cron.d/*
/etc/cron.deny
/etc/cron.daily/*
/etc/cron.hourly/*
/etc/cron.monthly/*
/etc/cron.weekly/*
/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/securetty
/etc/security/opasswd
/etc/shadow
/etc/ssh/sshd_config
/etc/stunnel/*
/etc/sysconfig/*
/etc/sysctl.conf
/etc/vsftpd.ftpusers
/etc/vsftpd/vsftpd.conf

/var/log/faillog
/var/log/lastlog
/var/spool/at/*
/var/spool/cron/tabs/*

```

Use the command `lastlog` to detect unusual patterns of logins.

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

```

atq
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 \( ! -uid 0 -o -perm +022 \)

```

## 5.2 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
----- -----	
<code>audit</code>	Directory for audit logs
<code>boot.msg</code>	Messages from system startup
<code>lastlog</code>	Last successful log in (see <code>lastlog(8)</code> )
<code>vsftpd.log</code>	Transaction log of the VSFTP daemon
<code>localmessages</code>	Written by <code>syslog</code>
<code>mail</code>	Written by <code>syslog</code> , contains messages from the MTA ( <code>postfix</code> )
<code>messages</code>	Written by <code>syslog</code> , contains messages from <code>su</code> and <code>ssh</code>
<code>news/</code>	<code>syslog</code> news entries (not used in the evaluated configuration)
<code>warn</code>	Written by <code>syslog</code>
<code>wtmp</code>	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.

## 5.3 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.

### 5.3.1 Intended usage of the audit subsystem

The Controlled Access Protection Profile (CAPP) specifies the auditing capabilities that a compliant system must support. The evaluated configuration described here is based on these requirements.

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

CAPP 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 open 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.

### 5.3.2 Selecting the events to be audited

You MAY make changes to the set of system calls and events that are to be audited. CAPP 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 a suggested audit configuration for CAPP systems in the `/usr/share/doc/audit-*/capp.rules` file. It contains 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 this file to `/etc/audit.rules` and modify the configuration according to your local requirements, including the option of using an empty audit rules file to disable auditing if not required.

You MAY selectively disable and enable auditing for specific events or users as required by modifying the *audit.rules* file as documented in the *auditctl*(8) man page. It is RECOMMENDED that you always reconfigure the audit system by modifying the `/etc/audit.rules` file and then running the following command to reload the audit rules:

```
auditctl -R /etc/audit.rules
```

This procedure ensures that the state of the audit system always matches the content of the `/etc/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.

### 5.3.3 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 by process ID
ausearch -p 3000

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

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

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.

### 5.3.4 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.

### 5.3.5 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/auditd.conf` file. You **MAY** change the `auditd.conf` file to suit your local requirements.

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** make the following change to the `/etc/rc.d/init.d/single` script to help ensure user processes are terminated. Add the lines marked with a "+" sign at the start of the line to the file at the indicated location, but do not include the "+" sign itself. You do not need to include the comment.

```
--- /etc/rc.d/init.d/single.orig      2006-06-15 17:45:37.000000000 -0400
+++ /etc/rc.d/init.d/single          2006-06-15 18:25:03.000000000 -0400
@@ -40,6 +40,20 @@
     $i start
done

+echo "Killing everything... "
+
+# Tell kernel to kill all processes except init and current process
+# (see kill(2) man page). The kernel locks the task list while
+# killing processes which prevents new processes being created
+# during the operation, and does not permit forking processes that
+# have a kill signal pending, so this should be reliable. Send
+# multiple signals for extra paranoia anyway.
+#
+# See also the "Robin Hood and Friar Tuck" hack description:
+#   http://catb.org/jargon/html/meaning-of-hack.html
+#
+kill -KILL -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
+
+# Now go to the single user level.
+echo $"Telling INIT to go to single user mode."
+exec init -t1 S
```

Halting the system is **RECOMMENDED** and most certain way to ensure all user processes are stopped. The following settings are **RECOMMENDED** in the `/etc/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.

### 5.3.6 Reliability of audit data

*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/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.

## 5.4 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.

# 6 Security guidelines for users

## 6.1 Online 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:

```
less /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.

## 6.2 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 may be available. 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* may not be due to having been infected with malicious code. Always remember that the device you are typing your password into has the ability to save and re-use your authentication information, so you are in effect giving the computer you are using the right to do any and all actions in your name. Insecure handling of authentication information is the leading

cause for exploits of otherwise secure systems, and SSH can only protect the information during transit, and offers no protection at all against an insecure end point.

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

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

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

### 6.3 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 will be prompted 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.
- Use either a password containing at least one character each from all four character classes; or a passphrase containing 12 total characters from any three of the four character classes. The character classes are defined as follows:

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

- 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'9twiciSd' too weak; contained in RHEL documentation"
=> P'9twiciRd
```

## 6.4 Access control for files and directories

Linux is a multiuser operating system. 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).

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.

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).

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.

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.

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.

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.

## 6.5 Data import / export

The system comes with various tools to archive data (*tar*, *star*, *cpio*). If ACLs are used, then only *star* **MUST** be used to handle the files and directories as the other commands do not support ACLs. The options *-H=exustar -acl* must be used with *star*.

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

# 7 Appendix

## 7.1 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.

"Red Hat Enterprise Linux 4 Installation Guide for the x86, Itanium and AMD64 Architectures",

<http://www.redhat.com/docs/manuals/enterprise/RHEL-4-Manual/x8664-multi-install-guide/>

"Red Hat Enterprise Linux 4 System Administration Guide",

<http://www.redhat.com/docs/manuals/enterprise/RHEL-4-Manual/sysadmin-guide/>

"Red Hat Enterprise Linux 4 Reference Guide",

<http://www.redhat.com/docs/manuals/enterprise/RHEL-4-Manual/ref-guide/>

"Red Hat Enterprise Linux 4 Security Guide",

<http://www.redhat.com/docs/manuals/enterprise/RHEL-4-Manual/security-guide/>

David A. Wheeler, "Secure Programming for Linux and Unix HOWTO",

[/usr/share/doc/howto/en/html\\_single/Secure-Programs-HOWTO.html](/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](/usr/share/doc/howto/en/html_single/Security-HOWTO.html)  
<http://www.linuxsecurity.com/docs/LDP/Security-HOWTO/>

## 7.2 Literature

Ellen Siever, Stephen Spainhour, Stephen Figgins, & Jessica P. Hekman, "Linux in a Nutshell, 3rd Edition", O'Reilly 2000, ISBN 0596000251

Simson Garfinkel, Gene Spafford, Alan Schwartz, "Practical Unix & Internet Security, 3rd Edition", O'Reilly 2003, ISBN 0596003234

Aleen Frisch, "Essential System Administration, 3rd Edition", O'Reilly 2002, ISBN 0596003439

Daniel J. Barrett, Richard Silverman, "SSH, The Secure Shell: The Definitive Guide", O'Reilly 2001, ISBN 0596000111

David N. Blank-Edelman, "Perl for System Administration", O'Reilly 2000, ISBN 1565926099

Shelley Powers, Jerry Peek, Tim O'Reilly, Mike Loukides, "Unix Power Tools, 3rd Edition", O'Reilly 2002, ISBN 0596003307

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

Linda Mui, "When You Can't Find Your UNIX System Administrator", O'Reilly 1995, ISBN 1565921046