Red Hat Linux 7.1

The Official Red Hat Linux Reference Guide

ISBN: N/A


Red Hat, Inc.

2600 Meridian Parkway
Durham, NC 27713 USA
+1 919 547 0012 (Voice)
+1 919 547 0024 (FAX)
888 733 4281 (Voice)
P.O. Box 13588
Research Triangle Park, NC 27709 USA

rhl-rg(EN)-7.1-Print-RHI (2001-03-09T14:50-0500)

Printed in Canada, Ireland, and Japan

Contents

Red Hat Linux 7.1

Introduction

Welcome to the *Official Red Hat Linux Reference Guide*.

The *Official Red Hat Linux Reference Guide* contains useful information about your Red Hat Linux system. From fundamental concepts, such as the structure of the Red Hat Linux filesystem, to the finer points of disk partitioning and authentication control, we hope you will find this book to be a valuable resource.

This guide is for you if you want to learn a bit more about how your Red Hat Linux system works. Topics that you will explore include the following:

- *Partitioning concepts* — An introduction to disk partitions and the techniques for placing more than one operating system on a single hard drive.
- *Booting Red Hat Linux* — Information about runlevels, `rc.d` directories, and how to start your favorite applications at boot time.
- *System and network security* — Find out the most common methods used by attackers to compromise your system and how to prevent security problems.
- *RAID concepts* — Making several disk drives act as a single logical unit, for increased performance and reliability.
- *Secure web server installation* — Adding encryption capabilities to your Apache web server.

Before reading this guide, you should be familiar with the contents of the *Official Red Hat Linux x86 Installation Guide* concerning installation issues, the *Official Red Hat Linux Getting Started Guide* for basic Linux concepts and the *Official Red Hat Linux Customization Guide* for general customization instructions. The *Official Red Hat Linux Reference Guide* contains information about advanced topics that may not affect every user, depending upon how they use their Red Hat Linux system.

HTML and PDF versions of all Official Red Hat Linux manuals are available online at http://www.redhat.com/support/manuals.

Finding Appropriate Documentation

You need documentation that is appropriate to your level of Linux expertise. Otherwise, you might feel overwhelmed or not find the necessary information to answer your questions. The *Official Red Hat Linux Reference Guide* deals with the more technical aspects and options of your Red Hat Linux system. This section will help you decide whether to look in this manual for the information you need or consider other Red Hat Linux manuals, including online sources, in your search.

Three different categories of people use Red Hat Linux, and each of these categories require different sets of documentation and informatative sources. To help you figure out where you should start, determine your own experience level:

New to Linux

This type of user has never used any Linux (or Linux-like) operating system before or has had only limited exposure to Linux. They may or may not have experience using other operating systems (such as Windows). Is this you? If so, skip ahead to *Documentation For First-Time Linux Users*.

Some Linux Experience

This type of user has installed and successfully used Linux (but not Red Hat Linux) before or may have equivalent experience with other Linux-like operating systems. Does this describe you? If so, turn to *For the More Experienced*.

Experienced User

This type of user has installed and successfully used Red Hat Linux before. If this describes you, turn to *Documentation for Linux Gurus*.

Documentation For First-Time Linux Users

For someone new to Linux, the amount of information available on any particular subject, such as printing, starting up the system or partitioning your hard drive, can be overwhelming. It helps to initially step back and gain a decent base of information centered around how Linux works before tackling these kinds of advanced issues.

Your first goal should be to obtain some useful documentation. This cannot be stressed enough. Without documentation, you will only become frustrated at your inability to get your Red Hat Linux system working the way you want.

You should acquire the following types of Linux documentation:

- *A brief history of Linux* — Many aspects of Linux are the way they are because of historical precedent. The Linux culture is also based on past events, needs or requirements. A basic understanding of the history of Linux will help you figure out how to solve many potential problems before you actually see them.
- *An explanation of how Linux works* — While delving into the most arcane aspects of the Linux kernel is not necessary, it is a good idea to know something about how Linux is put together. This is particularly important if you have been working with other operating systems, as some of the assumptions you currently hold about how computers work may not transfer from that operating system to Linux.

- *An introductory command overview (with examples)* — This is probably the most important thing to look for in Linux documentation. The underlying design philosophy for Linux is that it's better to use many small commands connected together in different ways than it is to have a few large (and complex) commands that do the whole job themselves. Without examples that illustrate this approach to doing things, you may find yourself intimidated by the sheer number of commands available on your Red Hat Linux system.

  Keep in mind that you do not have to memorize all of the available Linux commands. Different techniques exist to help you find the specific command you need to accomplish a task. You need only know the general way in which Linux functions, what you need to accomplish, and how to access the tool that will give you the exact instructions you need to execute the command.

The *Official Red Hat Linux x86 Installation Guide* is a excellent reference for helping you get your Red Hat Linux system successfully installed and initially configured. The *Official Red Hat Linux Getting Started Guide* covers the history of Linux, basic system commands, GNOME, KDE, RPM, and many other fundamental concepts. You should start with these two books and use them to build the base of your Red Hat Linux knowledge. Before long, more complicated concepts will begin to make sense because you already grasp the general ideas.

Beyond reading Red Hat Linux manuals, several other excellent documentation resources are available for little or no cost:

Introduction to Linux Websites

- http://www.redhat.com — On the Red Hat website, you will find links to the Linux Documentation Project (LDP), online versions of the Red Hat Linux manuals, FAQs (Frequently Asked Questions), a database which can help you find a Linux Users Group near you, technical information in the Red Hat Support Knowledge Base, and more.
- http://www.linuxheadquarters.com — The Linux Headquarters website features easy to follow, step-by-step guides for a variety of Linux tasks.

Introduction to Linux Newsgroups

You can participate in newsgroups by watching the discussions of others attempting to solve problems, or by actively asking or answering questions. Experienced Linux users are known to be extremely helpful when trying to assist new users with various Linux issues — especially if you are posing your questions in the right venue. If you do not have access to a news reader application, you can access this information via the web at http://www.deja.com. Dozens of Linux-related newsgroups exist, including the following:

- linux.help — A great place to get help from fellow Linux users.
- linux.redhat — This newsgroup primarily covers Red Hat Linux-specific issues.

- linux.redhat.install — Pose installation questions to this newsgroup or search it to see how others solved similar problems.
- linux.redhat.misc — Questions or requests for help that do not really fit into traditional categories go here.
- linux.redhat.rpm — A good place to go if you are having trouble using RPM to accomplish particular objectives.

Beginning Linux Books

- *Red Hat Linux for Dummies, 2nd Edition* by Jon "maddog" Hall; IDG
- *Special Edition Using Red Hat Linux* by Alan Simpson, John Ray and Neal Jamison; Que
- *Running Linux* by Matt Welsh and Lar Kaufman; O'Reilly & Associates
- *Red Hat Linux 7 Unleashed* by William Ball and David Pitts; Sams

The books suggested here are excellent primary sources of information for basic knowledge about a Red Hat Linux system. For more in-depth information concerning the various topics discussed throughout this book, many of the chapters list specific book titles, usually in an *Additional Resources* area.

For the More Experienced

If you have used other Linux distributions, you probably already have a basic grasp of the most frequently used commands. You may have installed your own Linux system, and maybe you have even downloaded and built software you found on the Internet. After installing Linux, however, configuration issues can be very confusing.

The *Official Red Hat Linux Customization Guide* is designed to help explain the various ways your Red Hat Linux system can be configured to meet specific objectives. Use this manual to learn about specific configuration options and how to put them into effect.

When you are installing software that is not covered in the *Official Red Hat Linux Customization Guide*, it is often helpful to see what other people in similar circumstances have done. HOWTO documents from the Linux Documentation Project, available at http://www.redhat.com/mirrors/LDP/HOWTO/HOWTO-INDEX/howtos.html, document particular aspects of Linux, from low-level kernel esoteric changes to using Linux for amateur radio station work.

Documentation for Linux Gurus

If you are a long-time Red Hat Linux user, you probably already know that one of the best ways to understand a particular program is to read its source code and/or configuration files. A major advantage of Red Hat Linux is the availability of the source code for anyone to read.

Obviously, not everyone is a C programmer, so the source code may not be helpful for you. However, if you have the knowledge and skills necessary to read it, the source code holds all of the answers.

Document Conventions

When you read this manual, you will see that certain words are represented in different fonts, typefaces, sizes and weights. This highlighting method is systematic; different words are represented in the same style to indicate their inclusion in a specific category. The types of words that are represented this way include the following:

`command`

Linux commands (and other operating system commands, when used) are represented this way. This style should indicate to you that you can type in the word or phrase on the command line and press [Enter] to invoke a command. Sometimes, a command contains words that would be displayed in a different style on their own (for example, filenames). In these cases, they are considered to be part of the command, so the entire phrase will be displayed as a command. For example:

Use the `cat testfile` command to view the contents of a file, named `testfile`, in the current working directory.

`filename`

Filenames, directory names, paths and RPM package names are represented this way. This style should indicate that a particular file or directory exists by that name on your Red Hat Linux system. For example:

The `.bashrc` file in your home directory contains bash shell definitions and aliases for your own use.

The `/etc/fstab` file contains information about different system devices and filesystems.

The `/usr/share/doc` directory contains documentation for various programs.

Install the `webalizer` RPM if you want to use a web server log file analysis program.

application

This style should indicate to you that the program named is an end-user application (as opposed to system software). For example:

Use Netscape Navigator to browse the web.

[key]

A key on the keyboard is shown in this style. For example:

To use [Tab] completion, type in a character and then press the [Tab] key. Your terminal will display the list of files in the directory that start with that letter.

[key]-[combination]

A combination of keystrokes is represented in this way. For example:

The [Ctrl]-[Alt]-[Backspace] key combination will restart the X Window System.

text found on a GUI interface

A title, word or phrase found on a GUI interface screen or window will be shown in this style. When you see text shown in this style, it is being used to identify a particular GUI screen or an element on a GUI screen (for example, text associated with a checkbox or field). For example:

On the GNOME **Control Center** screen, you can customize your GNOME window manager.

Select the **Require Password** checkbox if you'd like your screensaver to require a password before stopping.

top level of a menu on a GUI screen or window

When you see a word in this style, it indicates that the word is the top level of a pulldown menu. If you click on the word on the GUI screen, the rest of the menu should appear. For example:

Under **Settings** on a GNOME terminal, you'll see the following menu items: **Preferences**, **Reset Terminal**, **Reset and Clear**, and **Color selector**.

If you need to select a sequence of commands within a GUI menu, they will be shown like the following example:

Click on **Programs**=>**Applications**=>**Emacs** to start the Emacs text editor.

button on a GUI screen or window

This style indicates that the text will be found on a clickable button on a GUI screen. For example:

Click on the **Back** button to return to the web page you last viewed.

`computer output`

When you see text in this style, it indicates text displayed by the computer on the command line. You will see responses to commands you typed, error messages, and interactive prompts for your input during scripts or programs displayed in this way. For example:

Use the `ls` to display the contents of a directory:

```
$ ls
Desktop                 axhome          logs          nirvana.gif
Mail                    backupfiles     mail          reports
```

The output returned in response to the command (in this case, the contents of the directory) is shown in this style.

`prompt`

A prompt, which is a computer's way of signifying that it is ready for you to input something, will be shown in this style. Examples:

```
$
#
[truk@bleach truk]$
leopard login:
```

`user input`

Text that the user has to type, either on the command line or into a text box on a GUI screen, is displayed in this style. In the following example, **`text`** is displayed in this style:

To boot your system into the text based installation program, you will need to type in the **`text`** command at the `boot:` prompt.

Another example, with the word **`root`** displayed as something the user needs to type in:

If you need to log in as root when you first log into your system, and you're using the graphical login screen, at the `Login` prompt, type **`root`**. At the `Password` prompt, type in the root password.

glossary entry

A word that appears in the glossary will be shown in the body of the document in this style. For example:

The lpd **daemon** handles printing requests.

In this case, the style of the word **daemon** should indicate to you that a definition of the term is available in the glossary.

In addition, several different strategies are used throughout this manual to draw your attention to certain pieces of information. In order of how critical the information is to your system, these items will be marked as a note, a caution or a warning. For example:

Note

Remember that Linux is case sensitive. In other words, a rose is not a ROSE is not a rOsE.

Don’t do routine tasks as root — use a regular user account unless you need to use the root account to administer your system.

If you choose not to partition manually, a server-class installation will remove all existing partitions on all installed hard drives. Do not choose this installation class unless you are sure you have no data you need to save.

Using the Mouse

Red Hat Linux is designed to use a three-button mouse. If you have a two-button mouse, you should have selected three-button emulation during the installation process. If you’re using three-button emulation, pressing both mouse buttons at the same time equates to pressing the missing third (middle) button.

In this document, if you are instructed to click with the mouse on something, that means click the left mouse button. If you need to use the middle or right mouse button, that will be explicitly stated. (Of course, this will be reversed if you’ve configured your mouse to be used by a left handed person.)

The phrase "drag and drop" may be familiar to you. If you’re instructed to drag and drop an item on your GUI desktop, click on something and hold the mouse button down. While continuing to hold down the mouse button, drag the item by moving the mouse to a new location. When you’ve reached the desired location, release the mouse button to drop the item.

Copying and Pasting Text With X

Copying and pasting text is easy using your mouse and the X Window System. To copy text, simply click and drag your mouse over the text to highlight it. To paste the text somewhere, click the middle mouse button in the spot where the text should be placed.

More to Come

The *Official Red Hat Linux Reference Guide* is part of Red Hat's commitment to provide useful and timely support to Red Hat Linux users. Future editions will feature expanded information on changes to system structure and organization, new and powerful security tools, and other resources to help you extend the power of your Red Hat Linux system — and your ability to use it.

That's where you can help.

We Need Feedback!

If you find an error in the *Official Red Hat Linux Reference Guide*, or if you have thought of a way to make this manual better, we'd love to hear from you! Please submit a report in Bugzilla (http://bugzilla.redhat.com/bugzilla) against the component *Official Red Hat Linux Reference Guide*.

Be sure to mention the manual's identifier:

```
rhl-rg(EN)-7.1-Print-RHI (2001-03-09T14:50-0500)
```

If you mention the manual's identifier, we will know exactly which version of the guide you have.

If you have a suggestion for improving the documentation, try to be as specific as possible when describing it. If you have found an error, please include the section number and some of the surrounding text so we can find it easily.

Sign Up for Support

If you have an official edition of Red Hat Linux 7.1, please remember to sign up for the benefits you are entitled to as a Red Hat customer.

You may be entitled to any or all of the following benefits, depending upon the Official Red Hat Linux product you purchased:

- Official Red Hat support — Get help with your installation questions from Red Hat, Inc.'s support team.
- Red Hat Network — Easily update your packages and receive security notices that are customized for your system. Go to http://www.redhat.com/network for more details.
- Priority FTP access — No more late-night visits to congested mirror sites. Owners of Red Hat Linux 7.1 receive free access to priority.redhat.com, Red Hat's preferred customer FTP service, offering high bandwidth connections day and night.
- *Under the Brim: The Official Red Hat E-Newsletter* — Every month, get the latest news and product information directly from Red Hat.

To sign up for product support, go to http://www.redhat.com/apps/activate. You'll find your Product ID on a black, red and white card in your Official Red Hat Linux box.

To read more about technical support for Official Red Hat Linux, refer to the *Getting Technical Support* appendix in the *Official Red Hat Linux x86 Installation Guide*.

Good luck, and thank you for choosing Red Hat Linux!!

The Red Hat Documentation Team

Part I System-Related Reference

1 Filesystem Structure

1.1 Why Share a Common Structure?

An operating system's filesystem structure is its most basic level of organization. Almost all of the ways an operating system interacts with its users, applications, and security model are dependent upon the way it stores its files on a primary storage device (normally a hard disk drive). It is crucial for a variety of reasons that users, as well as programs at the time of installation and beyond, be able to refer to a common guideline to know where to read and write their binary, configuration, log, and other necessary files.

A filesystem can be seen in terms of two different logical categories of files:

- Shareable vs. unshareable files
- Variable vs. static files

Shareable files are those that can be accessed by various hosts; **unshareable** files are not available to any other hosts. **Variable** files can change at any time without system administrator intervention (whether active or passive); **static** files, such as documentation and binaries, do not change without an action from the system administrator or an agent that the system administrator has placed in motion to accomplish that task.

The reason for looking at files in this way has to do with the type of permissions given to the directory that holds them. The way in which the operating system and its users need to utilize the files determines the directory where those files should be placed, whether the directory is mounted read-only or read-write, and the level of access allowed on each file. The top level of this organization is crucial, as the access to the underlying directories can be restricted or security problems may manifest themselves if the top level is left disorganized or without a widely-utilized structure.

However, simply having a structure does not mean very much unless it is a standard. Competing structures can actually cause more problems than they fix. Because of this, Red Hat has chosen the the most widely-used filesystem structure and extended it only slightly to accommodate special files used within Red Hat Linux.

1.2 Overview of Filesystem Hierarchy Standard (FHS)

Red Hat is committed to the **Filesystem Hierarchy Standard** (**FHS**), a collaborative document that defines the names and locations of many files and directories. We will continue to track and follow the standard to keep Red Hat Linux FHS-compliant.

The current FHS document is the authoritative reference to any FHS-compliant filesystem, but the standard leaves many areas undefined or extensible. In this section, we provide an overview of the standard and a description of the parts of the filesystem not covered by the standard.

The complete standard is available at:

http://www.pathname.com/fhs

Compliance with the standard means many things, but the two most important are compatibility with other compliant systems and the ability to mount the `/usr` partition as read-only (because it contains common executables and is not meant to be changed by users). Since `/usr` can be mounted read-only, `/usr` can be mounted from the CD-ROM or from another machine via read-only NFS.

1.2.1 FHS Organization

The directories and files noted here are a small subset of those specified by the FHS document. Check the latest FHS document for the most complete information.

The `/dev` Directory

The `/dev` directory contains filesystem entries which represent devices that are attached to the system. These files are essential for the system to function properly.

The `/etc` Directory

The `/etc` directory is reserved for configuration files that are local to your machine. No binaries are to be put in `/etc`. Any binaries that were formerly put in `/etc` should now go into `/sbin` or possibly `/bin`.

The `X11` and `skel` directories should be subdirectories of `/etc`:

```
/etc
  |- X11
  |- skel
```

The `X11` directory is for X11 configuration files such as `XF86Config`. The `skel` directory is for "skeleton" user files, which are used to populate a home directory when a user is first created.

The `/lib` Directory

The `/lib` directory should contain only those libraries that are needed to execute the binaries in `/bin` and `/sbin`. These shared library images are particularly important for booting the system and executing commands within the root filesystem.

The `/mnt` Directory

The `/mnt` directory refers to temporarily mounted filesystems, such as CD-ROMs and floppy disks.

The /opt Directory

The /opt directory provides an area for usually large, static application software packages to be stored.

For packages that wish to avoid putting their files throughout the filesystem, /opt provides a logical and predictable organizational system under that package's directory. This gives the system administrator an easy way to determine the role of each file within a particular package.

For example, if sample is the name of a particular software package located within /opt, then all of its files could be placed within directories inside /opt/sample, such as /opt/sample/bin for binaries and /opt/sample/man for manual pages.

Large packages that encompass many different sub-packages, each of which accomplish a particular task, also go within /opt, giving that large package a standardized way to organize itself. In this way, our sample package may have different tools that each go in their own sub-directories, such as /opt/sample/tool1 and /opt/sample/tool2, each of which can have their own bin, man, and other similar directories.

The /sbin Directory

The /sbin directory is for executables used only by the root user. The executables in /sbin are only used to boot and mount /usr and perform system recovery operations. The FHS says:

> "/sbin typically contains files essential for booting the system in addition to the binaries in /bin. Anything executed after /usr is known to be mounted (when there are no problems) should be placed in /usr/sbin. Local-only system administration binaries should be placed into /usr/local/sbin."

At a minimum, the following programs should be in /sbin:

```
arp, clock, getty, halt, init, fdisk,
fsck.*, ifconfig, lilo, mkfs.*, mkswap, reboot,
route, shutdown, swapoff, swapon, update
```

The /usr Directory

The /usr directory is for files that can be shared across a whole site. The /usr directory usually has its own partition, and it should be mountable read-only. The following directories should be sub-directories of /usr:

```
/usr
  |- bin
  |- doc
  |- etc
  |- games
  |- include
```

```
|- kerberos
|- lib
|- libexec
|- local
|- man
|- sbin
|- share
|- src
|- X11R6
```

The `bin` directory contains executables, `doc` contains non-FHS compliant documentation pages, `etc` contains system-wide configuration files, `games` is for games, `include` contains C header files, `kerberos` contains binaries and much more for Kerberos, and `lib` contains object files and libraries that are not designed to be directly utilized by users or shell scripts. The `libexec` directory contains small helper programs called by other programs, `sbin` is for system administration binaries (those that do not belong in `/sbin`), `share` contains files that aren't architecture-specific, `src` is for source code, and `X11R6` is for the X Window System (`XFree86` on Red Hat Linux).

The `/usr/local` Directory

The FHS says:

> "The `/usr/local` hierarchy is for use by the system administrator when installing software locally. It needs to be safe from being overwritten when the system software is updated. It may be used for programs and data that are shareable amongst a group of hosts, but not found in `/usr`."

The `/usr/local` directory is similar in structure to the `/usr` directory. It has the following subdirectories, which are similar in purpose to those in the `/usr` directory:

```
/usr/local
       |- bin
       |- doc
       |- etc
       |- games
       |- info
       |- lib
       |- man
       |- sbin
       |- src
```

The `/var` Directory

Since the FHS requires that you be able to mount `/usr` read-only, any programs that write log files or need `spool` or `lock` directories should probably write them to the `/var` directory. The FHS states `/var` is for:

> "…variable data files. This includes spool directories and files, administrative and logging data, and transient and temporary files."

The following directories should be subdirectories of `/var`:

```
/var
  |- arpwatrch
  |- cache
  |- db
  |- ftp
  |- gdm
  |- kerberos
  |- lib
  |- local
  |- lock
  |- log
  |- named
  |- nis
  |- opt
  |- preserve
  |- run
  +- spool
       |- anacron
       |- at
       |- cron
       |- fax
       |- lpd
       |- mail
       |- mqueue
       |- news
       |- rwho
       |- samba
       |- slrnpull
       |- squid
       |- up2date
       |- uucp
       |- uucppublic
       |- vbox
       |- voice
  |- tmp
  |- www
  |- yp
```

System log files such as `messages` and `lastlog` go in `/var/log`. The `/var/lib/rpm` directory also contains the RPM system databases. Lock files go in `/var/lock`, usually in directories

particular for the program using the file. The /var/spool directory has subdirectories for various systems that need to store data files.

1.2.2 /usr/local in Red Hat Linux

In Red Hat Linux, the intended use for /usr/local is slightly different from that specified by the FHS. The FHS says that /usr/local should be where you store software that is to remain safe from system software upgrades. Since system upgrades from Red Hat are done safely with the RPM and Gnome-RPM, you don't need to protect files by putting them in /usr/local. Instead, we recommend you use /usr/local for software that is local to your machine.

For instance, let's say you have mounted /usr via read-only NFS from a host named jake. If there is a package or program you would like to install, but you are not allowed to write to jake, you should install it under /usr/local. Later perhaps, if you have managed to convince the system administrator of jake to install the program on /usr, you can uninstall it from /usr/local.

1.3 /proc and Its "Files"

The /proc directory contains special "files" that either extract information from or send information to the kernel.

However, the /proc directory is much more powerful than you might initially think. Through the various "files" in this directory (which are really not files at all but interfaces into the kernel), a system administrator can use /proc as an easy method of accessing information about the state of the kernel, the attributes of the machine, the states of individual processes, and more. By using cat in combination with the interfaces within /proc, you can immediately access an enormous amount of information about any system. As an example, if you want to see how the memory registers are currently assigned on your computer:

```
[truk@tictactoe /proc]$ cat iomem
00000000-0009fbff : System RAM
0009fc00-0009ffff : reserved
000a0000-000bffff : Video RAM area
000c0000-000c7fff : Video ROM
000f0000-000fffff : System ROM
00100000-07ffffff : System RAM
  00100000-002553d7 : Kernel code
  002553d8-0026d91b : Kernel data
e0000000-e3ffffff : VIA Technologies, Inc. VT82C597 [Apollo VP3]
e4000000-e7ffffff : PCI Bus #01
  e4000000-e4003fff : Matrox Graphics, Inc. MGA G200 AGP
  e5000000-e57fffff : Matrox Graphics, Inc. MGA G200 AGP
e8000000-e8ffffff : PCI Bus #01
  e8000000-e8ffffff : Matrox Graphics, Inc. MGA G200 AGP
```

```
ea000000-ea00007f : Digital Equipment Corporation DECchip 21140
  ea000000-ea00007f : eth0
ffff0000-ffffffff : reserved
[truk@tictactoe /proc]$
```

Or (and more usefully), if were connecting to an unknown machine and wanted to know its CPU type and speed, you can use the following command:

```
cat /proc/cpuinfo
```

Other valuable bits of system information can be gathered from `cmdline`, `meminfo`, `partitions`, and `version`, among others.

The directories in `/proc` symbolize a collection of information about a particular application or process. For example, the `/proc/sys/kernel` directory is full of information about the kernel, such as the maximum number of threads (`threads-max`) and the maximum number of messages (`msgmax`).

1.4 Special Red Hat Linux File Locations

In addition to the files pertaining to RPM that reside in `/var/lib/rpm` (see the RPM chapter in the *Official Red Hat Linux Customization Guide* for more information on RPM), there are two other special locations reserved for Red Hat Linux configuration and operation.

The configuration tools provided with Red Hat Linux install many scripts, bitmap, and text files in `/usr/lib/rhs`. Since these files are generated by software on your system, you probably won't want to edit any of them by hand.

The other special location (`/etc/sysconfig`) stores configuration information. Many scripts that run at boot time use the files in this directory. These files can be edited by hand, but they can also be configured using Linuxconf, a control panel tool, or another configuration tool. See *Official Red Hat Linux Customization Guide* for instructions on using Linuxconf.

2 Users and Groups

The control of **users** and **groups** exists at the core of Red Hat Linux system administration.

Users can be either actual people (accounts tied to a particular physical user) or logical users (accounts that exist for applications so that they can do particular things). Both types of users, actual or logical, have a **User ID** and **Group ID**. User IDs are usually unique (but don't have to be).

Groups are always logical expressions of organization. Users make up groups, and groups form the foundation of tying users together and giving them permissions to read, write, or execute a given file.

Any file created is assigned a user and group when it is made, as well as being assigned separate read, write, and execute permissions for the file's owner, the group assigned to the file, and any other users on that host. The user and group of a particular file, as well as the permissions on that file, can be changed by root or, to a lesser extent, by the creator of the file.

Proper management of users and groups, as well as assigning and revoking permissions, is one of the most important tasks of any system administrator. Thankfully, Red Hat Linux makes this job as easy as possible while preserving the security of the files on the host.

2.1 Tools for User and Group Administration

Managing users and groups has traditionally been tedious, but Red Hat Linux provides a few tools and conventions to make users and groups easier to manage.

While you can use `useradd` to create a new user from the shell prompt, a popular way to manage users and groups is through Linuxconf (see the *Official Red Hat Linux Customization Guide* for details on Linuxconf).

2.2 Standard Users

In Table 2–1, *Standard Users*, you'll find the standard users set up by the installation process (this is essentially the `/etc/passwd` file). The Group ID (GID) in this table is the *primary group* for the user. See Section 2.4, *User Private Groups* for details on how groups are used.

Table 2–1 Standard Users

User	UID	GID	Home Directory	Shell
root	0	0	`/root`	`/bin/bash`
bin	1	1	`/bin`	
daemon	2	2	`/sbin`	

User	UID	GID	Home Directory	Shell
adm	3	4	/var/adm	
lp	4	7	/var/spool/lpd	
sync	5	0	/sbin	/bin/sync
shutdown	6	0	/sbin	/sbin/shutdown
halt	7	0	/sbin	/sbin/halt
mail	8	12	/var/spool/mail	
news	9	13	/var/spool/news	
uucp	10	14	/var/spool/uucp	
operator	11	0	/root	
games	12	100	/usr/games	
gopher	13	30	/usr/lib/gopher-data	
ftp	14	50	/var/ftp	
nobody	99	99	/	

2.3 Standard Groups

In Table 2–2, *Standard Groups*, you'll find the standard groups as set up by the installation process (this is essentially the `/etc/group` file).

Table 2–2 Standard Groups

Group	GID	Members
root	0	root
bin	1	root, bin, daemon
daemon	2	root, bin, daemon
sys	3	root, bin, adm
adm	4	root, adm, daemon
tty	5	

Group	GID	Members
disk	6	root
lp	7	daemon, lp
mem	8	
kmem	9	
wheel	10	root
mail	12	mail
news	13	news
uucp	14	uucp
man	15	
games	20	
gopher	30	
dip	40	
ftp	50	
nobody	99	
users	100	

2.4 User Private Groups

Red Hat Linux uses a **user private group** (**UPG**) scheme, which makes UNIX groups much easier to use. The UPG scheme does not add or change anything in the standard UNIX way of handling groups; it simply offers a new convention. Whenever you create a new user, by default, he or she has a unique group. The scheme works as follows:

User Private Group

Every user has a primary group; the user is the only member of that group.

umask = 002

Traditionally, on UNIX systems the umask is 022, which prevents other users *and other members of a user's primary group* from modifying a user's files. Since every user has his or her own private group in the UPG scheme, this "group protection" is not needed. A umask of 002 will prevent users from modifying other users' private files. The umask is set in `/etc/profile`.

setgid bit on Directories

If you set the setgid bit on a directory (with `chmod g+s` *`directory`*), files created in that directory will have their group set to the directory's group.

Many IT organizations like to create a group for each major project and then assign people to the group if they need to access that group's files. Using this traditional scheme, managing files has been difficult because when someone creates a file, it is associated with the primary group to which he or she belongs. When a single person works on multiple projects, it is difficult to associate the right files with the right group. Using the UPG scheme, however, groups are automatically assigned to files created within that directory, which makes managing group projects that share a common directory very simple.

For example, let's say you have a big project called *devel*, with many people editing the *devel* files in a `devel` directory. Make a group called `devel`, `chgrp` the `devel` directory to `devel`, and add all of the *devel* users to the `devel` group.

You can add a user to a group using Linuxconf (see the *Official Red Hat Linux Customization Guide*). If you prefer to use the command line, use the `/usr/sbin/groupadd` *`groupname`* command to create a group. The `/usr/bin/gpasswd -a` *`loginname groupname`* command will add a user *loginname* to a group. (See the `groupadd` and `gpasswd` man pages if you need more information on their options.) The `/etc/group` file contains the group information for your system.

If you created the `devel` group, added users to the `devel` group, changed the group for `devel` directory to the `devel` group, and set the setgid bit for the `devel` directory, all *devel* users will be able to edit the *devel* files and create new files in the `devel` directory. The files they create will always retain their `devel` group status, so other *devel* users will always be able to edit them.

If you have multiple projects like *devel* and users who are working on multiple projects, these users will never have to change their umask or group when they move from project to project. If set correctly, the setgid bit on each project's main directory "selects" the proper group for all files created in that directory.

Since each user's home directory is owned by the user and their private group, it is safe to set the setgid bit on the home directory. However, by default, files are created with the primary group of the user, so the setgid bit would be redundant.

2.4.1 User Private Group Rationale

Although UPG has existed in Red Hat Linux for quite some time, many people still have questions about it, such as why UPG is necessary. Consider the following rationale for the scheme:

- You would like to have a group of people work on a set of files in the `/usr/lib/emacs/site-lisp` directory. You trust a few people to modify the directory but certainly not everyone.
- So, first you create an `emacs` group:

```
/usr/sbin/groupadd emacs
```

Next, you enter:

```
chown -R root.emacs /usr/lib/emacs/site-lisp
```

to associate the contents of the directory with the `emacs` group and add the proper users to the group:

```
/usr/bin/gpasswd -a <username> emacs
```

- To allow the users to actually create files in the directory you enter:

  ```
  chmod 775 /usr/lib/emacs/site-lisp
  ```

- But when a user creates a new file it is assigned the group of the user's default group (usually `users`). To prevent this you enter:

  ```
  chmod 2775 /usr/lib/emacs/site-lisp
  ```

  which causes everything in the directory to be created with the `emacs` group.

- But the new file needs to be mode 664 for another user in the emacs group to be able to edit it. To do this you make the default umask 002.

- Well, this all works fine, except that if your default group is `users`, every file you create in your home directory will be writable by everybody in `users` (usually everyone).

- To fix this, you make each user have a "private group" as their default group.

At this point, by making the default umask 002 and giving everyone a private default group, you can easily set up groups that users can take advantage of without any extra work every time users write files to the group's common directory. Just create the group, add the users, and do the above `chown` and `chmod` on the group's directories.

3 Boot Process, Init, and Shutdown

This chapter contains information on what happens when you boot or shut down your Red Hat Linux system.

3.1 Introduction

One of the most powerful aspects of Red Hat Linux concerns its open method of starting and stopping the operating system, where it loads specified programs using their particular configurations, permits you to change those configurations to control the boot process, and shuts down in a graceful and organized way. While other operating systems attempt to control the way the computer boots or prevent you from customizing what happens at shutdown, Red Hat Linux allows you full access to every step in the process.

Beyond the question of controlling of the boot or shutdown process, the open nature of Red Hat Linux makes it much easier to determine the exact source of most problems associated with starting up or shutting down your system. An understanding of this process is quite beneficial for even basic troubleshooting.

3.2 Behind the Scenes of the Boot Process

Note

This section looks at the x86 boot process, in particular. Depending on your system's architecture, your boot process may be slightly different. However, once the kernel is found and loaded by the system, the default Red Hat Linux boot process is identical across all architectures. Please see Section 3.8, *Differences in the Boot Process of Other Architectures* for more information on a non-x86 boot process.

When a computer is booted, the processor looks at the end of the system memory for the **BIOS** (Basic Input/Output System) and runs it. The BIOS program is written into read-only permanent memory and is always available for use. The BIOS provides the lowest level interface to peripheral devices and controls the first step of the boot process.

The BIOS tests the system, looks for and checks peripherals, and then looks for a drive to use to boot the system. Usually, it checks the floppy drive (or CD-ROM drive on many newer systems) for bootable media, if present, and then it looks to the hard drive. The order of the drives used for booting is usually controlled by a particular BIOS setting on the system. Once Red Hat Linux is installed on

a hard drive of a system, the BIOS looks for a **Master Boot Record** (MBR) starting at the first sector on the first hard drive, loads its contents into memory, and passes control to it.

This MBR code then looks for the first active partition and reads the partition's boot record. The boot record contains instructions on how to load the boot loader, **LILO** (*LI*nux *LO*ader). The MBR then loads LILO, which takes over the process (if LILO is installed in the MBR). In the default Red Hat Linux configuration, LILO uses the settings in the MBR to display boot options and allow for user input on which operating system to actually start up.

But this begs the question: How does LILO in the MBR know what to do when the MBR is read? LILO actually has already written the instructions there through the use of `lilo` with the `/etc/lilo.conf` configuration file.

3.2.1 Options in `/etc/lilo.conf`

Most of the time, you will have no need to change the Master Boot Record on your hard drive unless you need to boot a newly installed operating system or are looking to use a new kernel. If you do need to create a new MBR using LILO but using a different configuration, you will need edit `/etc/lilo.conf` and run `lilo` again.

WARNING

If you are planning to edit `/etc/lilo.conf`, be sure to make a backup copy of the file before making any changes. Also, be sure that you have a working boot floppy available so that you will be able to boot the system and make changes to the MBR if there is a problem. See the man pages for `mkbootdisk` for more information on creating a boot disk.

The file `/etc/lilo.conf` is used by `lilo` to determine which operating system(s) to utilize or which kernel to start, as well as to know where to install itself (for example, `/dev/hda` for the first IDE hard drive). A sample `/etc/lilo.conf` file looks like this:

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
message=/boot/message
lba32
default=linux

image=/boot/vmlinuz-2.4.0-0.43.6
```

```
    label=linux
    initrd=/boot/initrd-2.4.0-0.43.6.img
    read-only
    root=/dev/hda5

  other=/dev/hda1
    label=dos
```

This example shows a system configured to boot two operating systems: Red Hat Linux and DOS. Here is a deeper look at a few of the lines of this file (your `/etc/lilo.conf` may look a little different):

- `boot=/dev/hda` tells LILO to look on the first hard disk on the first IDE controller.
- `map=/boot/map` locates the map file. In normal use, this should not be modified.
- `install=/boot/boot.b` tells LILO to install the specified file as the new boot sector. In normal use, this should not be altered. If the `install` line is missing, LILO will assume a default of `/boot/boot.b` as the file to be used.
- The existence of `prompt` tells LILO to show you whatever is referenced in the `message` line. While it is not recommended that you remove the `prompt` line, if you do remove it, you can still get a prompt by holding down the [Shift] key while your machine starts to boot.
- `timeout=50` sets the amount of time that LILO will wait for user input before proceeding with booting the `default` line entry. This is measured in tenths of a second, with 50 as the default.
- `message=/boot/message` refers to the screen that LILO displays to let you select the operating system or kernel to boot.
- `lba32` describes the hard disk geometry to LILO. Another common entry here is `linear`. You should not change this line unless you are very aware of what you are doing. Otherwise, you could put your system in a state where it cannot boot.
- `default=linux` refers to the default operating system for LILO to boot from the options listed below this line. The name `linux` refers to the `label` line below in each of the boot options.
- `image=/boot/vmlinuz-2.4.0-0.43.6` specifies the linux kernel to boot with this particular boot option.
- `label=linux` names the operating system option in the LILO screen. In this case, it also is the name that is referred to by the `default` line.
- `initrd=/boot/initrd-2.4.0-0.43.6.img` refers to the **initial ram disk** image that is used at boot time to actually initialize and start the devices that makes booting the kernel possible. The initial ram disk is a collection of machine-specific drivers necessary to operate the hard drive and anything needed to load the kernel. You should never try to share initial ram disks between machines unless they are identical in their hardware configurations (and even then, it is a bad idea).

- `read-only` specifies that the root partition (see the `root` line below) as one that cannot be changed, only read.
- `root=/dev/hda5` tells LILO what disk partition to use as the root partition.

LILO then shows the Red Hat Linux initial screen with the different operating systems or kernels it has been configured to boot. If you only have Red Hat Linux installed and have not changed anything in `/etc/lilo.conf`, you will only see linux as an option. If you have set up LILO to boot other operating systems as well, this screen is your chance to select what operating system will boot. Use your arrow keys to highlight the operating system and press [Enter]

If you would like to have a command prompt to enter commands to LILO, press [Cntl]-[X]. LILO displays a `LILO:` prompt on the screen and waits for a preset period of time for input from the user. (The amount of time LILO waits is set by the `timeout` line in the `/etc/lilo.conf` file.) If your `/etc/lilo.conf` is set to give LILO a choice of operating systems, at this time you could type in the label for whichever operating system you want to boot.

If LILO is booting Linux, it first loads the kernel into memory, which is a `vmlinuz` file (plus a version number, for example, `vmlinuz-2.4.0-xx`) located in the `/boot` directory. Then the kernel passes control to `init`.

At this point, with the kernel loaded into memory and operational, Linux is already started, although at a very basic level. However, with no applications utilizing the kernel and with no ability for the user to provide meaningful input to the system, not much can be done with it. The `init` program solves this problem by bringing up the various services that allow the system to perform its particular role.

3.2.2 Init

The kernel finds `init` in `/sbin` and executes it, and `init` which coordinates the rest of the boot process.

When `init` starts, it becomes the parent or grandparent of all of the processes that start up automatically on your Red Hat Linux system. First, it runs the `/etc/rc.d/rc.sysinit` script, which sets your path, starts swapping, checks the filesystems, and so on. Basically, `rc.sysinit` takes care of everything that your system needs to have done at system initialization. For example, on a networked system, `rc.sysinit` uses the information in the `/etc/sysconfig/network` file to initialize network processes. Most systems use a clock, so on them `rc.sysinit` uses the `/etc/sysconfig/clock` file to initialize the clock. If you have special serial port processes that need to be initialized, `rc.sysinit` may also run `rc.serial`.

Then, `init` runs the `/etc/inittab` script, which describes how the system should be set up in each **runlevel** and sets the default runlevel. (See Section 3.4, *Init Runlevels* for more information on init runlevels.) This file states, among other things, that `/sbin/update` should be run whenever a runlevel starts. The `update` program is used to flush dirty buffers back to disk.

Whenever the runlevel changes, `init` uses the scripts in `/etc/rc.d/init.d` to start and stop various services, such as your web server, DNS server, and so on. First, `init` sets the source function library for the system (commonly `/etc/rc.d/init.d/functions`), which spells out how to start or kill a program and how to find out the PID of a program. Then, `init` determines the current and the previous runlevel.

Next, `init` starts all of the background processes necessary for the system to run by looking in the appropriate `rc` directory for that runlevel (`/etc/rc.d/rc`*<x>*`.d`, where the *<x>* is numbered 0-6). `init` runs each of the kill scripts (their file name starts with a `K`) with a `stop` parameter. Then, `init` runs all of the start scripts (their file names start with an `S`) in the appropriate runlevel directory with a `start` so that all services and applications are started correctly. In fact, you can execute these same scripts manually after the system is finished booting with a command like `/etc/rc.d/init.d/httpd stop` or `service httpd stop` logged in as root. This will stop the `httpd` server.

Note

When starting services manually, you should be root. If you get a error when executing `service httpd stop`, you may not have `/sbin` pathed in `/root/.bashrc` (or the correct `.rc` file for your preferred shell). You can either type the full command of `/sbin/service httpd stop` or add `export PATH="$PATH:/sbin"` to your shell `.rc` file. If you edit your shell configuration file, log out and back in as root to make the changed shell configuration file take effect.

None of the scripts that actually start and stop the services are located in `/etc/rc.d/rc`*<x>*`.d`. Rather, all of the files in `/etc/rc.d/rc`*<x>*`.d` are **symbolic links** that point to actual scripts located in `/etc/rc.d/init.d`. A symbolic link is nothing more than a file that simply points to another file, and they are used in this case because they can be created and deleted without affecting the actual script that kills or starts the service. The symbolic links to the various scripts are numbered in a particular order so that they start in that order. You can change the order in which the services start up or are killed by changing the name of the symbolic link that refers to the script that actually starts or kills the service. You can give symbolic links the same number as other symbolic links if you want that service start or stop right before or after another service.

For example, for runlevel 5, `init` looks into the `/etc/rc.d/rc5.d` directory and might finds the following (your system and configuration may vary):

```
K01pppoe -> ../init.d/pppoe
K05innd -> ../init.d/innd
K10ntpd -> ../init.d/ntpd
K15httpd -> ../init.d/httpd
```

```
K15mysqld -> ../init.d/mysqld
K15pvmd -> ../init.d/pvmd
K16rarpd -> ../init.d/rarpd
K20bootparamd -> ../init.d/bootparamd
K20nfs -> ../init.d/nfs
K20rstatd -> ../init.d/rstatd
K20rusersd -> ../init.d/rusersd
K20rwalld -> ../init.d/rwalld
K20rwhod -> ../init.d/rwhod
K25squid -> ../init.d/squid
K28amd -> ../init.d/amd
K30mcserv -> ../init.d/mcserv
K34yppasswdd -> ../init.d/yppasswdd
K35dhcpd -> ../init.d/dhcpd
K35smb -> ../init.d/smb
K35vncserver -> ../init.d/vncserver
K45arpwatch -> ../init.d/arpwatch
K45named -> ../init.d/named
K50snmpd -> ../init.d/snmpd
K54pxe -> ../init.d/pxe
K55routed -> ../init.d/routed
K60mars-nwe -> ../init.d/mars-nwe
K61ldap -> ../init.d/ldap
K65kadmin -> ../init.d/kadmin
K65kprop -> ../init.d/kprop
K65krb524 -> ../init.d/krb524
K65krb5kdc -> ../init.d/krb5kdc
K75gated -> ../init.d/gated
K80nscd -> ../init.d/nscd
K84ypserv -> ../init.d/ypserv
K90ups -> ../init.d/ups
K96irda -> ../init.d/irda
S05kudzu -> ../init.d/kudzu
S06reconfig -> ../init.d/reconfig
S08ipchains -> ../init.d/ipchains
S10network -> ../init.d/network
S12syslog -> ../init.d/syslog
S13portmap -> ../init.d/portmap
S14nfslock -> ../init.d/nfslock
S18autofs -> ../init.d/autofs
S20random -> ../init.d/random
S25netfs -> ../init.d/netfs
S26apmd -> ../init.d/apmd
S35identd -> ../init.d/identd
S40atd -> ../init.d/atd
```

```
S45pcmcia -> ../init.d/pcmcia
S55sshd -> ../init.d/sshd
S56rawdevices -> ../init.d/rawdevices
S56xinetd -> ../init.d/xinetd
S60lpd -> ../init.d/lpd
S75keytable -> ../init.d/keytable
S80isdn -> ../init.d/isdn
S80sendmail -> ../init.d/sendmail
S85gpm -> ../init.d/gpm
S90canna -> ../init.d/canna
S90crond -> ../init.d/crond
S90FreeWnn -> ../init.d/FreeWnn
S90xfs -> ../init.d/xfs
S95anacron -> ../init.d/anacron
S97rhnsd -> ../init.d/rhnsd
S99linuxconf -> ../init.d/linuxconf
S99local -> ../rc.local
```

These symbolic links tell `init` that it needs to kill `pppoe`, `innd`, `ntpd`, `httpd`, `mysqld`, `pvmd`, `rarpd`, `bootparamd`, `nfs`, `rstatd`, `rusersd`, `rwalld`, `rwhod`, `squid`, `amd`, `mcserv`, `yppasswdd`, `dhcpd`, `smb`, `vncserver`, `arpwatch`, `named`, `snmpd`, `pxe`, `routed`, `mars-nwe`, `ldap`, `kadmin`, `kprop`, `krb524`, `krb5kdc`, `gated`, `nscd`, `ypserv`, `ups`, and `irda`. After all processes are killed, `init` looks into the same directory and finds start scripts for `kudzu`, `reconfig`, `ipchains`, `portmap`, `nfslock`, `autofs`, `random`, `netfs`, `apmd`, `identd`, `atd`, `pcmcia`, `sshd`, `rawdevices`, `xinetd`, `lpd`, `keytable`, `isdn`, `sendmail`, `gpm`, `canna`, `crond`, `FreeWnn`, `xfs`, `anacron`, `rhnsd`, and `linuxconf`. The last thing `init` does is run `/etc/rc.d/rc.local` to run any special scripts configured for that host. At this point, the system is considered to be operating at runlevel 5.

After `init` has progressed through all of the runlevels, the `/etc/inittab` script forks a `getty` process for each virtual console (login prompts) for each runlevel (runlevels 2-5 get all six; runlevel 1, which is single user mode, only gets one console; runlevels 0 and 6 get no virtual consoles). Basically, `getty` opens tty lines, sets their modes, prints the login prompt, gets the user's name, and then initiates a login process for that user. This allows users to authenticate themselves to the system and begin to use it.

Also, `/etc/inittab` tells `init` how it should handle a user hitting [Ctrl]-[Alt]-[Delete] at the console. As Red Hat Linux should be properly shut down and restarted rather immediately power-cycled, `init` is told to execute the command `/sbin/shutdown -t3 -r now` when a user hits those keys. In addition, `/etc/inittab` states what `init` should do in case of power failures, if your system has a UPS unit attached to it.

In runlevel 5, `/etc/inittab` runs a script called `/etc/X11/prefdm`. The `prefdm` script runs the preferred X display manager (`gdm` if you're running GNOME, `kdm` if you're running KDE, or

`xdm` if you're running AnotherLevel) based on the contents of the `/etc/sysconfig/desktop` directory.

At this point, you should be looking at a login prompt. All that, and it only took a few seconds.

3.2.3 SysV Init

As we have seen, the `init` program is run by the kernel at boot time. It is in charge of starting all the normal processes that need to start up with the system. These include the getty processes that allow you to log in, NFS daemons, FTP daemons, and anything else you want to run when your machine boots.

SysV init is the standard init process in the Linux world to control the startup of software at boot time, because it is easier to use and more powerful and flexible than the traditional BSD `init`.

SysV init also differs from BSD init in that the configuration files are in `/etc/rc.d` instead of residing directly in `/etc`. In `/etc/rc.d`, you will find `rc`, `rc.local`, `rc.sysinit`, and the following directories:

```
init.d
rc0.d
rc1.d
rc2.d
rc3.d
rc4.d
rc5.d
rc6.d
```

SysV init represents each of the init runlevels with a separate directory, using `init` and symbolic links in each of the directories to actually stop and start the services as the system moves from runlevel to runlevel.

In summary, the chain of events for a SysV init boot is as follows:

- The kernel looks in `/sbin` for `init`
- `init` runs the `/etc/rc.d/rc.sysinit` script
- `rc.sysinit` handles most of the boot loader's processes and then runs `rc.serial` (if it exists)
- `init` runs all the scripts for the default runlevel
- `init` runs `/etc/rc.d/rc.local`

The default runlevel is decided in `/etc/inittab`. You should have a line close to the top like:

```
id:3:initdefault:
```

The default runlevel is 3 in this example, the number after the first colon. If you want to change it, you can edit `/etc/inittab` by hand. Be very careful when you are editing the `inittab` file. If you do mess up, you can fix it by rebooting, accessing the `boot:` prompt with [Cntl]-[X], and typing:

```
boot:  linux single
```

This *should* allow you to boot into single-user mode so you can re-edit `inittab` to its previous value.

Next, we'll discuss information in the files within `/etc/sysconfig` that define the parameters used by different system services when they start up.

3.3 Sysconfig Information

The following information outlines some of the various files in `/etc/sysconfig`, their function, and their contents. This information is not intended to be complete, as many of these files have a variety of options that are only used in very specific or rare circumstances.

3.3.1 Files in `/etc/sysconfig`

The following files are normally found in `/etc/sysconfig`:

- `amd`
- `apmd`
- `authconfig`
- `cipe`
- `clock`
- `desktop`
- `firewall`
- `harddisks`
- `hwconf`
- `i18n`
- `init`
- `irda`
- `keyboard`
- `kudzu`
- `mouse`

- network
- pcmcia
- rawdevices
- sendmail
- soundcard
- ups
- vncservers

It is possible that your system may be missing a few of them if the corresponding program that would need that file is not installed.

Let's take a look at each one.

/etc/sysconfig/amd

The /etc/sysconfig/amd file contains various parameters used by amd allowing for the auto-mounting and automatic unmounting of filesystems.

/etc/sysconfig/apmd

The /etc/sysconfig/apmd file is used by apmd as a configuration for what things to start/stop/change on suspend or resume. It is set up to turn on or off apmd during startup, depending on whether your hardware supports **Advanced Power Management** (**APM**) or if you choose not to use it. apm is a monitoring daemon that works with power management code within the Linux kernel. It can alert you to a low battery if you are using Red Hat Linux on a laptop, among other things.

/etc/sysconfig/authconfig

The /etc/sysconfig/authconfig file sets the kind of authorization to be used on the host. It contains one or more of the following lines:

- USEMD5=*<value>*, where *<value>* is one of the following:
  - yes — MD5 is used for authentication.
  - no — MD5 is not used for authentication.

- USEKERBEROS=*<value>*, where *<value>* is one of the following:
  - yes — Kerberos is used for authentication.
  - no — Kerberos is not used for authentication.

- USELDAPAUTH=<*value*>, where <*value*> is one of the following:
  - yes — LDAP is used for authentication.
  - no — LDAP is not used for authentication.

/etc/sysconfig/cipe

The /etc/sysconfig/cipe file configures cipe when it starts.

It may contains the following sample values:

- DEVICE=eth0, which specifies the network adapter that cipe will utilize.
- PORT=9999, which designates the UDP port number to be used by the cipe process in both endpoints.
- PEER=0.0.0.0, which specifies the real address of the remote cipe endpoint. You can set this address dynamically by setting this value to 0.0.0.0.
- IPADDR=0.0.0.0, which specifies the virtual address at the local end of the cipe tunnel.
- PTPADDR=0.0.0.0, which specifies the virtual address at the remote end of the cipe tunnel.

/etc/sysconfig/clock

The /etc/sysconfig/clock file controls the interpretation of values read from the system clock. Earlier releases of Red Hat Linux used the following values (which are deprecated):

- CLOCKMODE=<*value*>, where <*value*> is one of the following:
  - GMT — Indicates that the clock is set to Universal Time (Greenwich Mean Time).
  - ARC — Indicates the ARC console's 42-year time offset is in effect (for Alpha-based systems only).

Currently, the correct values are:

- UTC=<*value*>, where <*value*> is one of the following boolean values:
  - true — Indicates that the clock is set to Universal Time. Any other value indicates that it is set to local time.
- ARC=<*value*>, where <*value*> is the following:
  - true — Indicates the ARC console's 42-year time offset is in effect. Any other value indicates that the normal UNIX epoch is assumed (for Alpha-based systems only).

- ZONE=*<filename>* — Indicates the timezone file under /usr/share/zoneinfo that /etc/localtime is a copy of, such as:

```
ZONE="America/New York"
```

/etc/sysconfig/desktop

The /etc/sysconfig/desktop file specifies the desktop manager to be run, such as:

```
DESKTOP="GNOME"
```

/etc/sysconfig/firewall

The /etc/sysconfig/firewall file contains various firewall settings. By default, this file is created but empty.

/etc/sysconfig/harddisks

The /etc/sysconfig/harddisks file allows you to tune your hard drive(s).

WARNING

Don't make changes to this file lightly. If you change the default values stored here, you could corrupt all of the data on your hard drive(s).

The /etc/sysconfig/harddisks file may contain the following:

- USE_DMA=1, where setting this to 1 enables DMA. However, with some chipsets and hard drive combinations, DMA can cause data corruption. *Check with your hard drive documentation or manufacturer before enabling this.*
- Multiple_IO=16, where a setting of 16 allows for multiple sectors per I/O interrupt. When enabled, this feature reduces operating system overhead by 30-50%. *Use with caution.*
- EIDE_32BIT=3 enables (E)IDE 32-bit I/O support to an interface card.
- LOOKAHEAD=1 enables drive read-lookahead.
- EXTRA_PARAMS= specifies where extra parameters can be added.

/etc/sysconfig/hwconf

The /etc/sysconfig/hwconf file lists all the hardware that kudzu detected on your system, as well as the drivers used, vendor ID and device ID information. The kudzu program detects and

configures new and/or changed hardware on a system. The `/etc/sysconfig/hwconf` file is not meant to be manually edited. If you do edit it, devices could suddenly show up as being added or removed.

`/etc/sysconfig/i18n`

The `/etc/sysconfig/i18n` file sets the default language, such as:

```
LANG="en_US"
```

`/etc/sysconfig/init`

The `/etc/sysconfig/init` file controls how the system will appear and function during bootup.

The following values may be used:

- `BOOTUP=<value>`, where `<value>` is one of the following:
  - `BOOTUP=color` means the standard color boot display, where the success or failure of devices and services starting up is shown in different colors.
  - `BOOTUP=verbose` means an old style display, which provides more information than purely a message of success or failure.
  - Anything else means a new display, but without ANSI-formatting.

- `RES_COL=<value>`, where `<value>` is the number of the column of the screen to start status labels. Defaults to 60.
- `MOVE_TO_COL=<value>`, where `<value>` moves the cursor to the value in the `RES_COL` line. Defaults to ANSI sequences output by echo `-e`.
- `SETCOLOR_SUCCESS=<value>`, where `<value>` sets the color to a color indicating success. Defaults to ANSI sequences output by echo `-e`, setting the color to green.
- `SETCOLOR_FAILURE=<value>`, where `<value>` sets the color to a color indicating failure. Defaults to ANSI sequences output by echo `-e`, setting the color to red.
- `SETCOLOR_WARNING=<value>`, where `<value>` sets the color to a color indicating warning. Defaults to ANSI sequences output by echo `-e`, setting the color to yellow.
- `SETCOLOR_NORMAL=<value>`, where `<value>` sets the color to 'normal'. Defaults to ANSI sequences output by echo `-e`.
- `LOGLEVEL=<value>`, where `<value>` sets the initial console logging level for the kernel. The default is 7; 8 means everything (including debugging); 1 means nothing except kernel panics. `syslogd` will override this once it starts.

- PROMPT=<*value*>, where <*value*> is one of the following boolean values:
  - yes — Enables the key check for interactive mode.
  - no — Disables the key check for interactive mode.

/etc/sysconfig/irda

The /etc/sysconfig/irda file controls how infared devices on your system are configured at startup.

The following values may be used:

- IRDA=<*value*>, where <*value*> is one of the following boolean values:
  - yes — irattach will be run, which periodically checks to see if anything is trying to connect to the infrared port, such as another notebook computer trying to make a network connection. For infrared devices to work on your system, this line must be set to yes.
  - no — irattach will not be run, preventing infrared device communication.

- DEVICE=<*value*>, where <*value*> is the device (usually a serial port) that handles infrared connections.
- DONGLE=<*value*>, where <*value*> specifies the type of dongle being used for infrared communication. This setting exists for people who use serial dongles rather than real infrared ports. A dongle is a device that is attached to a traditional serial port to communicate via infrared. This line is commented out by default because notebooks with real infrared ports are far more common than computers with add-on dongles.
- DISCOVERY=<*value*>, where <*value*> is one of the following boolean values:
  - yes — Starts irattach in discovery mode, meaning it actively checks for other infrared devices. This needs to be turned on for the machine to be actively looking for an infrared connection (meaning the peer that does not initiate the connection).
  - no — Does not start irattach in discovery mode.

/etc/sysconfig/keyboard

The /etc/sysconfig/keyboard file controls the behavior of the keyboard. The following values may be used:

- KEYBOARDTYPE=sun|pc, which is used on SPARCs only. sun means a Sun keyboard is attached on /dev/kbd, and pc means a PS/2 keyboard connected to a PS/2 port.
- KEYTABLE=*<file>*, where *<file>* is the name of a keytable file. For example: KEYTABLE="us". The files that can be used as keytables start in /usr/lib/kbd/keymaps/i386 and branch into different keyboard layouts from there, all labeled *<file>*.kmap.gz. The first file found beneath /usr/lib/kbd/keymaps/i386that matches the KEYTABLE setting is used.

/etc/sysconfig/kudzu

The /etc/sysconfig/kuzdu allows you to specify a safe probe of your system's hardware by kudzu at boot time. A safe probe is one that disables serial port and DDC monitor probing.

- SAFE=*<value>*, where *<value>* is one of the following:
  - yes — kuzdu does a safe probe.
  - no — kuzdu does a normal probe.

/etc/sysconfig/mouse

The /etc/sysconfig/mouse file is used to specify information about the available mouse. The following values may be used:

- FULLNAME=*<value>*, where *<value>* refers to the full name of the kind of mouse being used.
- MOUSETYPE=*<value>*, where *<value>* is one of the following:
  - microsoft — A Microsoft™ mouse.
  - mouseman — A MouseMan™ mouse.
  - mousesystems — A Mouse Systems™ mouse.
  - ps/2 — A PS/2 mouse.
  - msbm — A Microsoft™ bus mouse.
  - logibm — A Logitech™ bus mouse.
  - atibm — An ATI™ bus mouse.
  - logitech — A Logitech™ mouse.
  - mmseries — An older MouseMan™ mouse.
  - mmhittab — An mmhittab mouse.

- XEMU3=*<value>*, where *<value>* is one of the following boolean values:
  - yes — The mouse only has two buttons, but three mouse buttons should be emulated.
  - no — The mouse already has three buttons.

- XMOUSETYPE=*<value>*, where *<value>* refers to the kind of mouse used when X is running. The options here are the same as the MOUSETYPE setting in this same file.

In addition, /dev/mouse is a symbolic link that points to the actual mouse device.

/etc/sysconfig/network

The /etc/sysconfig/network file is used to specify information about the desired network configuration. The following values may be used:

- NETWORKING=*<value>*, where *<value>* is one of the following boolean values:
  - yes — Networking should be configured.
  - no — Networking should not be configured.

- HOSTNAME=*<value>*, where *<value>* should be the **Fully Qualified Domain Name** (**FQDN**), such as hostname.domain.com, but can be whatever hostname you want.

Note

For compatibility with older software that people might install (such as trn), the /etc/HOSTNAME file should contain the same value as here.

- GATEWAY=*<value>*, where *<value>* is the IP address of the network's gateway.
- GATEWAYDEV=*<value>*, where *<value>* is the gateway device, such as eth0.
- NISDOMAIN=*<value>*, where *<value>* is the NIS domain name.

/etc/sysconfig/pcmcia

The /etc/sysconfig/pcmcia file is used to specify PCMCIA configuration information. The following values may be used:

- PCMCIA=*<value>*, where *<value>* is one of the following:
  - yes — PCMCIA support should be enabled.

  - `no` — PCMCIA support should not be enabled.

- `PCIC=<value>`, where `<value>` is one of the following:
  - `i82365` — The computer has an i82365-style PCMCIA socket chipset.
  - `tcic` — The computer has a tcic-style PCMCIA socket chipset.

- `PCIC_OPTS=<value>`, where `<value>` is the socket driver (i82365 or tcic) timing parameters.
- `CORE_OPTS=<value>`, where `<value>` is the list of `pcmcia_core` options.
- `CARDMGR_OPTS=<value>`, where `<value>` is the list of options for the PCMCIA `cardmgr` (such as `-q` for quiet mode; `-m` to look for loadable kernel modules in the specified directory, and so on). Read the `cardmgr` man page for more information.

/etc/sysconfig/rawdevices

The `/etc/sysconfig/rawdevices` file is used to configure raw device bindings, such as:

```
/dev/raw/raw1 /dev/sda1
/dev/raw/raw2 8 5
```

/etc/sysconfig/sendmail

The `/etc/sysconfig/sendmail` file allows messages to be sent to one or more recipients, routing the message over whatever networks are necessary. The file sets the default values for the Sendmail application to run. Its default values are to run as a background daemon, and to check its queue once an hour in case something has backed up.

The following values may be used:

- `DAEMON=<value>`, where `<value>` is one of the following boolean values:
  - `yes` — Sendmail should be configured to listen to port 25 for incoming mail. `yes` implies the use of Sendmail's `-bd` options.
  - `no` — Sendmail should not be configured to listen to port 25 for incoming mail.

- `QUEUE=1h` which is given to Sendmail as `-q$QUEUE`. The `-q` option is not given to Sendmail if `/etc/sysconfig/sendmail` exists and `QUEUE` is empty or undefined.

/etc/sysconfig/soundcard

The /etc/sysconfig/soundcard file is generated by sndconfig and should not be modified. The sole use of this file is to determine what card entry in the menu to pop up by default the next time sndconfig is run. Soundcard configuration information is located in the /etc/modules.conf file.

It may contain the following:

- CARDTYPE=<*value*>, where <*value*> is set to, for example, SB16 for a Soundblaster 16 sound card.

/etc/sysconfig/ups

The /etc/sysconfig/ups file is used to specify information about any **Uninterruptible Power Supplies** (**UPS**) connected to your system. A UPS can be very valuable for a Red Hat Linux system because it gives you time to correctly shut down the system in the case of power interruption. The following values may be used:

- SERVER=<*value*>, where <*value*> is one of the following:
  - yes — A UPS device is connected to your system.
  - no — A UPS device is not connected to your system.

- MODEL=<*value*>, where <*value*> must be one of the following or set to NONE if no UPS is connected to the system:
  - apcsmart — For a APC SmartUPS™ or similar device.
  - fentonups — For a Fenton UPS™.
  - optiups — For an OPTI-UPS™ device.
  - bestups — For a Best Power™ UPS.
  - genericups — For a generic brand UPS.
  - ups-trust425+625 — For a Trust™ UPS.

- DEVICE=<*value*>, where <*value*> specifies where the UPS is connected, such as /dev/ttyS0.
- OPTIONS=<*value*>, where <*value*> is a special command that needs to be passed to the UPS.

/etc/sysconfig/vncservers

The /etc/sysconfig/vncservers file configures how the **Virtual Network Computing** (**VNC**) server starts up. VNC is a remote display system which allows you to view a desktop environment not only on the machine where it is running but across different networks (from a LAN to the Internet) and using a wide variety of machine architectures.

It may contain the following:

- VNCSERVERS=*<value>*, where *<value>* is set to something like "1:root".

3.3.2 Files in /etc/sysconfig/network-scripts/

The following files are normally found in /etc/sysconfig/network-scripts, where *<if-name>* represents the name of the network interface:

- /etc/sysconfig/network-scripts/ifup
- /etc/sysconfig/network-scripts/ifdown
- /etc/sysconfig/network-scripts/network-functions
- /etc/sysconfig/network-scripts/ifcfg-*<if-name>*
- /etc/sysconfig/network-scripts/ifcfg-*<if-name>*-*<clone-name>*
- /etc/sysconfig/network-scripts/chat-*<if-name>*
- /etc/sysconfig/network-scripts/dip-*<if-name>*
- /etc/sysconfig/network-scripts/ifup-post

Let's take a look at each one.

/etc/sysconfig/network-scripts/ifup and /etc/sysconfig/network-scripts/ifdown

These are symbolic links to /sbin/ifup and /sbin/ifdown, respectively. These are the only two scripts in this directory that should be called directly; these two scripts call all the other scripts as needed. These symbolic links are here for legacy purposes only — they will probably be removed in future versions, so only /sbin/ifup and /sbin/ifdown should currently be used.

These scripts normally take one argument: the name of the device (such as eth0). They are called with a second argument of boot during the boot sequence so that devices that are not meant to be brought up on boot (ONBOOT=no, [see below]) can be ignored at that time.

/etc/sysconfig/network-scripts/network-functions

Not really a public file. Contains functions which the scripts use for bringing interfaces up and down. In particular, it contains most of the code for handling alternative interface configurations and interface change notification through `netreport`, which is the program that tells network management scripts to send a SIGIO signal to the process which called `netreport` when any network interface status changes occur.

/etc/sysconfig/network-scripts/ifcfg-*<if-name>* and /etc/sysconfig/network-scripts/ifcfg-*<if-name>*:*<clone-name>*

The first file defines an interface, while the second file contains only the parts of the definition that are different in a "alias" (or alternative) interface. Both require that an *<if-name>* (name of a network interface) is specified. For example, the network numbers might be different but everything else might be the same, so only the network numbers would be in the clone file while all the device information would be in the base `ifcfg` file.

The items that can be defined in an `ifcfg` file depend on the interface type.

The following values are common:

- `DEVICE=`*<name>*, where *<name* is the name of the physical device (except dynamically-allocated PPP devices where it is the "logical name").
- `IPADDR=`*<addr>*, where *<addr>* is the IP address.
- `NETMASK=`*<mask>*, where *<mask>* is the netmask value.
- `NETWORK=`*<addr>*, where *<addr>* is the network address.
- `BROADCAST=`*<addr>*, where *<addr>* is the broadcast address.
- `GATEWAY=`*<addr>*, where *<addr>* is the gateway address.
- `ONBOOT=`*<answer>*, where *<answer>* is one of the following:
  - `yes` — This device should be activated at boot-time.
  - `no` — This device should not be activated at boot-time.
- `USERCTL=`*<answer>*, where *<answer>* is one of the following:
  - `yes` — Non-root users are allowed to control this device.
  - `no` — Non-root users are not allowed to control this device.
- `BOOTPROTO=`*<proto>*, where *<proto>* is one of the following:
  - `none` — No boot-time protocol should be used.
  - `bootp` — The BOOTP protocol should be used.

  - `dhcp` — The DHCP protocol should be used.

The following values are common to all SLIP files:

- `PERSIST=<answer>`, where *`<answer>`* is one of the following:
  - `yes` — This device should be kept active at all times, even if deactivated after a modem hang up.
  - `no` — This device should not be kept active at all times.

- `MODEMPORT=<port>`, where *`<port>`* is the modem port's device name (for example, "`/dev/modem`").
- `LINESPEED=<baud>`, where *`<baud>`* is the modem's linespeed (for example, "`115200`").
- `DEFABORT=<answer>`, where *`<answer>`* is one of the following:
  - `yes` — Insert default abort strings when creating/editing the script for this interface.
  - `no` — Do not insert default abort strings when creating/editing the script for this interface.

`/etc/sysconfig/network-scripts/chat-<if-name>`

This file is a chat script for SLIP connections and is intended to establish the connection. For SLIP devices, a DIP script is written from the chat script.

`/etc/sysconfig/network-scripts/ifup-post`

This file is called when any network device (except a SLIP device) comes up. It calls `/etc/sysconfig/network-scripts/ifup-routes` to bring up static routes that depend on that device, brings up aliases for that device, and sets the hostname if it is not already set — and a hostname can be found for the IP for that device. `ifup-post` sends SIGIO to any programs that have requested notification of network events.

This file could be extended to set up name service configuration, call arbitrary scripts, and more, as needed.

3.4 Init Runlevels

The idea behind operating different services at different runlevels essentially revolves around the fact that different systems can be used in a different ways. Some services cannot be used until the system is in a particular state, or **mode**, such as ready for more than one user or has networking available. There are times in which you may want to operate the system at a lower mode, such as testing a networking

problem in runlevel 2 or leaving the system in runlevel 3 without an X session running. In these cases, running services that depend upon a higher system mode to function doesn't make sense because they won't work correctly anyway. By already having each service assigned to start when its particular runlevel is reached, you ensure an orderly start up process and can quickly change the mode of the machine without worrying about which services to manually start or stop.

Generally, Red Hat Linux operates in runlevel 3 — full multi-user mode. The following runlevels are defined in Red Hat Linux:

- `0` — Halt
- `1` — Single-user mode
- `2` — Multi-user mode, without networking
- `3` — Full multi-user mode
- `4` — Not used
- `5` — Full multi-user mode (with an X-based login screen)
- `6` — Reboot

The default runlevel for a system to boot to and stop is configured in `/etc/inittab`. For more information on `/etc/inittab`, see Section 3.2.3, *SysV Init*.

If your machine gets into a state where it will not boot due to a bad `/etc/inittab` or will not let you log in because you have a corrupted `/etc/passwd` (or if you have simply forgotten your password), boot into single-user mode by typing **`linux single`** at the LILO `boot:` prompt. A very bare system will boot, and you will have a command shell from which you can fix things.

3.5 Initscript Utilities

The `chkconfig` utility in `/sbin` provides a simple command-line tool for maintaining the `/etc/rc.d/init.d` directory hierarchy. It relieves system administrators from having to directly manipulate the numerous symbolic links in the directories under `/etc/rc.d`.

In addition, there is the `ntsysv` utility in `/usr/sbin` that provides a screen-oriented interface, which you may find easier to use than `chkconfig`'s command-line interface. Both of these utilities should be run as root.

Please see the `chkconfig` and `ntsysv` man pages for more information.

3.6 Running Programs at Boot Time

The file `/etc/rc.d/rc.local` script is run by `init` at boot time, after all other initialization is complete, and whenever you change runlevels. You can add additional initialization commands here. For instance, you may want to start up additional daemons or initialize a printer.

In addition, if you require serial port setup, you can create and edit `/etc/rc.serial`, and it will be executed automatically at boot time. This script can run a number of `setserial` commands to specially configure the system's serial ports. See the `setserial` man page for more information.

The default `/etc/rc.d/rc.local` simply creates a login banner with your kernel version and machine type.

3.7 Shutting Down

To shut down Red Hat Linux, issue the `shutdown` command. You can read the `shutdown` man page for complete details, but the two most common uses are:

```
/sbin/shutdown -h now
/sbin/shutdown -r now
```

You must run `shutdown` as root. After shutting everything down, the `-h` option will halt the machine, and the `-r` option will reboot.

Although the `reboot` and `halt` commands are now able to invoke `shutdown` if run while the system is in runlevels 1-5, it is a bad habit to get into, as not all Linux-like operating systems have this feature.

WARNING

If your computer does not power itself down, you should not turn off the computer until you see a message indicating that the system is halted or finished shutting down.

Failure to wait for this message will mean that you may be turning off the machine before your hard drive partitions are unmounted. This can cause filesystem corruption, even to the point where your system may not boot the next time it attempts to start up. Be patient when halting your system.

3.8 Differences in the Boot Process of Other Architectures

Each computer architecture supported by Red Hat Linux boots the operating system in a different way. However, once the Red Hat Linux kernel begins booting and hands off the boot process to `init`, the same events happen on each architecture in exactly the same way. The only difference is in the way Red Hat Linux finds the kernel to load it in order to start `init`.

Consult the installation information for each of the architectures for detailed information about the different boot methods.

4 Lightweight Directory Access Protocol (LDAP)

4.1 What is LDAP?

LDAP (**Lightweight Directory Access Protocol**) is a proposed open standard for global or local directory services over a network and/or the Internet. A directory, in this sense, is very much like a phone book. LDAP can handle other information, but at present it is typically used to associate names with phone numbers and email addresses. LDAP directories are designed to support a high volume of queries, but the data stored in the directory doesn't change very often.

LDAP is much more useful than a paper phone book, because LDAP's design is intended to support propagation over LDAP servers throughout the Internet, much like the **Domain Name Service** (**DNS**). DNS servers help to connect computers to one another based on fully qualified domain names or the type of service requested from a domain, such as mail exchange. Without DNS servers, hostnames could not be translated into IP addresses, which are required for TCP/IP communication. In the future, LDAP could provide the same type of global access to many types of directory information. Currently, LDAP is more commonly used within a single large organization, like a college or a company, for directory services.

LDAP is a client-server system. An LDAP client connects to an LDAP server and either queries it for information or provides information that needs to be entered into the directory. The server either answers the query, refers the query to another LDAP server, or accepts the information for incorporation into the directory, based on the permission of the user.

LDAP is sometimes known as **X.500 Lite**. X.500 is an international standard for directories and full-featured, but it is also complex, requiring a lot of computing resources and the full OSI stack. LDAP, in contrast, can run easily on a PC and over TCP/IP. LDAP can access X.500 directories but does not support every capability of X.500.

This chapter will refer to the configuration and use of OpenLDAP, an open source implementation of LDAP. OpenLDAP includes `slapd` (a stand-alone LDAP server), `slurpd` (a stand-alone LDAP replication server), libraries implementing the LDAP protocol, utilities, tools, and sample clients.

4.2 Pros and Cons of LDAP

The main benefit of using LDAP is the consolidation of certain types of information within your organization. For example, all of the different lists of users within your organization can be merged into one LDAP directory. This directory can be queried by any LDAP-enabled applications that need this information. It can also be used by users who need directory information.

Other LDAP benefits include its ease of implementation (compared to X.500) and its well-defined Application Programming Interface (API), which means that the number of LDAP-enabled applications and LDAP gateways should increase in the future.

On the negative side, if you want to use LDAP, you will need LDAP-enabled applications or the ability to use LDAP gateways. While LDAP usage should only increase, currently there are not very many LDAP-enabled applications available for Linux. Also, while LDAP does support some access control, it does not possess as many security features as X.500.

4.3 Uses for LDAP

Several Netscape applications, including web browsers using the Netscape Roaming Access feature, are LDAP-enabled. Sendmail can use LDAP to look up addresses. Your organization can use LDAP as an organization-wide directory and/or name service (in place of NIS or flat files). You can even use a personal LDAP server to keep track of your own email address book (see Section 4.11, *Additional Resources*).

Since LDAP is an open and configurable protocol, it can be used to store almost any type of information relating to a particular organizational structure.

4.3.1 LDAP Applications

Several LDAP client applications are available that greatly simplify viewing and changing LDAP information:

- LDAP Browser/Editor — A user-friendly tool written in 100% Java for easy deployment across different platforms, available at http://www.iit.edu/~gawojar/ldap
- GQ — A GTK-based LDAP client, available with the Red Hat Linux 7.1 distribution or at http://biot.com/gq
- kldap — An LDAP client for the KDE Project, available at http://www.mountpoint.ch/oliver/kldap

4.3.2 LDAP and PAM

LDAP can be used as an authentication service via the `pam_ldap` module. LDAP is commonly used as a central authentication server so that users have a unified login that covers console logins, POP servers, IMAP servers, machines connected to the network using Samba, and even Windows NT/2000 machines. Using LDAP, all of these login situations can rely on the same user ID and password combination, greatly simplifying administration. The `pam_ldap` module is provided in the `nss_ldap` package.

4.4 LDAP Terminology

An **entry** is one unit in an LDAP directory. An entry is identified or referenced by its unique **Distinguished Name** (**DN**).

An entry has **attributes**, which are pieces of information directly associated with the entry. For example, an organization could be an LDAP entry. Attributes associated with the organization might be its fax number, its address, and so on. People can also be entries in the LDAP directory. Common attributes for people include their telephone numbers and their e-mail addresses.

Certain attributes are required, while other attributes are optional. An **objectclass** sets which attributes are required and which are optional. Objectclass definitions are found in various schema files, located in the `/etc/openldap/schema` directory.

The **LDAP Data Interchange Format** (LDIF) is an ASCII text format for LDAP entries. Files that import or export data to and from LDAP servers must be in LDIF format. An LDIF entry looks like this:

```
[<id>]
dn: <distinguished name>
<attrtype>: <attrvalue>
<attrtype>: <attrvalue>
<attrtype>: <attrvalue>
```

An entry can contain as many `<attrtype>: <attrvalue>` pairs as needed. A blank line indicates that the entry is finished and that another entry is about to begin.

Your `<attrtype>` and `<attrvalue>` pairs *must* be defined in a schema before they can be used. You cannot simply define them in an LDIF file and expect an LDAP server without corresponding data in its schema files to be able to use this information.

Everything enclosed within < > is variable and can be set by you when you add an LDAP entry, with the exception of the `<id>`. The `<id>` is a number normally set by the LDAP tools when you add an entry, and you will probably never need to manually set one.

4.5 OpenLDAP 2.0 Enhancements

OpenLDAP 2.0 represents a major upgrade for the application, bringing with it:

- *LDAPv3 Support* — Now works with SASL, TLS, and SSL, among other improvements, in full compliance with RFC 2251-2256; many of the changes since LDAPv2 are aimed to help make LDAP a much more secure protocol.
- *IPv6 Support* — Now supports the next generation Internet Protocol.
- *LDAP Over IPC* — OpenLDAP can communicate within a particular system without having to go over a network, making it more secure.
- *Updated C API* — Improves the way programmers can connect to and use the application.
- *LDIFv1 Support* — Full compliance with the LDAP Data Interchange Format (LDIF) version 1.
- *Enhanced Stand-Alone LDAP Server* — Includes an updated access control system, thread pooling, better tools and much more.

4.6 OpenLDAP Files

OpenLDAP configuration files are installed into the `/etc/openldap` directory. If you do an `ls` on `/etc/openldap`, you will see something like:

```
ldap.conf         ldapsearchprefs.conf  schema
ldapfilter.conf   ldaptemplates.conf    slapd.conf
```

4.6.1 Edit `/etc/openldap/slapd.conf`

The `slapd.conf` file, located in `/etc/openldap`, contains the configuration information needed by your `slapd` LDAP server. You will need to edit this file to make it specific to your domain and server.

The suffix line names the domain for which the LDAP server will provide information. The suffix line should be changed from:

```
suffix          "dc=your-domain, dc=com"
```

so that it reflects your domain name. For example:

```
suffix          "dc=acmewidgets, dc=com"
```

or

```
suffix          "dc=acmeuniversity, dc=edu"
```

The rootdn entry is the DN for a user who is unrestricted by the access control or administrative limit parameters set for operations on the LDAP directory. The rootdn user can be thought of as the root user for the LDAP directory. The rootdn line needs to be changed from:

```
rootdn          "cn=root, dc=your-domain, dc=com"
```

to something like:

```
rootdn          "cn=root, dc=redhat, dc=com"
```

or

```
rootdn          "cn=ldapmanager, dc=my_organization, dc=org"
```

Change the rootpw line from:

```
rootpw          secret
```

to something like

```
rootpw          {crypt}s4L9sOIJo4kBM
```

In the above example, you are using an encrypted root password, which is a much better idea than leaving a plain text root password in the `slapd.conf` file. To make this crypt string, you can use Perl:

```
perl -e "print crypt('passwd','a_salt_string');"
```

In the previous Perl line, *`salt_string`* is a two character salt, and *`passwd`* is the plain text version of the password.

You could also copy a `passwd` entry out of `/etc/passwd`, but this won't work if the `passwd` entry is an MD5 password (the default in Red Hat Linux 7.1).

4.6.2 The `schema` Directory

New to OpenLDAP version 2, the `schema` directory holds the various LDAP definitions, previously located in the `slapd.at.conf` and `slapd.oc.conf` files. All **attribute syntax definitions** and **objectclass definitions** are now located in the different schema files. The various schema files are referenced in `/etc/openldap/slapd.conf` using `include` lines, as shown in this example:

```
include  /etc/openldap/schema/core.schema
include  /etc/openldap/schema/cosine.schema
include  /etc/openldap/schema/inetorgperson.schema
include  /etc/openldap/schema/nis.schema
include  /etc/openldap/schema/rfc822-MailMember.schema
include  /etc/openldap/schema/autofs.schema
include  /etc/openldap/schema/kerberosobject.schema
```

You should not modify any of the schema items defined in the schema files installed by OpenLDAP.

You can extend the schema used by OpenLDAP to support additional attribute types and object classes using the default schema files as a guide. To do this, create a `local.schema` file in the `/etc/openldap/schema` directory. Reference this new schema within `slapd.conf` by adding the following line below your default `include` schema lines:

```
include  /etc/openldap/schema/local.schema
```

Next, go about defining your new attribute types and object classes within the `local.schema` file. Many organizations use existing attribute types and object classes from the schema files installed by default and modify them for use in the `local.schema` file. This can help you to learn the schema syntax while meeting the immediate needs of your organization.

Extending schemas to match certain specialized requirements is quite involved and beyond the scope of this chapter. Visit http://www.openldap.org/doc/admin/schema.html for information on writing new schema files.

4.7 OpenLDAP Daemons and Utilities

The OpenLDAP package includes two daemons: `slapd` and `slurpd`.

The `slapd` daemon is the stand-alone LDAP daemon, which you'll need to run to support LDAP.

The `slurpd` daemon controls the replication of LDAP directories over a network by sending changes from the master LDAP directory to slave LDAP directories. You won't need to run `slurpd` unless you have more than one LDAP server on your network. If you have two or more LDAP servers, `slurpd` will keep the various LDAP directories in sync.

OpenLDAP also includes some utilities in `/usr/bin` for adding, modifying and deleting entries in an LDAP directory:

- `ldapmodify` — Modify entries in an LDAP database, accepting input via a file or standard input.
- `ldapadd` — Adds entries to your directory, accepting input via a file or standard input; `ldapadd` is actually a hard link to `ldapmodify -a`.
- `ldapsearch` — Searches for entries in the LDAP directory using a shell prompt.

- `ldapdelete` — Deletes entries from an LDAP directory, accepting input via a file or a shell prompt.

With the exception of `ldapsearch`, each of these utilities is much more easily used by referencing a file with the changes to be made rather than typing the commands one after the other. Each of their respective man pages covers the syntax of these files.

To import or export blocks of information with a `slapd` directory or perform similar administrative tasks, different utilities, located in `/usr/sbin`, are required:

- `slapadd` — Adds entries from an LDIF file to an LDAP directory. For example, execute `/usr/sbin/slapadd -l` *`ldif`* where *`ldif`* is the name of the LDIF file containing the new entries.
- `slapcat` — Pulls entries out of an LDAP directory and saves them in an LDIF file. For example, execute `/usr/sbin/slapcat -l` *`ldif`* where *`ldif`* is the name of the target LDIF file to contain the entries from the LDAP directory.
- `slapindex` — Reindexes the `slapd` database based on the actual current database content. Execute `/usr/sbin/slapindex` to begin reindexing.
- `slappasswd` — Generates a user password value for use with `ldapmodify` or the `rootpw` value in `/etc/openldap/slapd.conf`. Execute `/usr/sbin/slappasswd` to create the password.

WARNING

Be sure to stop `slapd` before using `slapadd`, `slapcat` or `slapindex`. Otherwise, you are risking the consistency of your LDAP database.

See the man pages for each of these utilities for more information about how to use them.

4.8 Modules for Adding Extra Functionality to LDAP

Red Hat Linux includes several packages that add functionality to LDAP.

The `nss_ldap` module is an LDAP module for the **Solaris Nameservice Switch** (NSS). NSS is a set of C library extensions necessary for accessing LDAP directory information, instead of or in addition to the **Network Information Service** (NIS) name service and/or flat files. The `nss_ldap` module is needed to use LDAP as a native name service.

The `pam_ldap` module is needed to integrate LDAP authentication into the Pluggable Authentication Modules (PAM) API. If you use `pam_ldap`, users can authenticate and change their password

using LDAP directories. The nss_ldap and pam_ldap modules are provided in the nss_ldap package.

Red Hat Linux also includes LDAP modules for the Apache Web server. The auth_ldap module is for authenticating HTTP clients against the user entries in an LDAP directory. The php-ldap module adds LDAP support to the PHP4 HTML-embedded scripting language. The auth_ldap and php-ldap modules must be compiled into Apache as **Dynamic Shared Objects** (**DSO**s) in order to work.

4.9 LDAP How To: A Quick Overview

This section provides a quick overview of the steps you will need to take to get an LDAP directory working.

1. Make sure the openldap RPM, and any other LDAP-related RPMs that you need, are installed.
2. Refer to either the Quick Start Guide at the OpenLDAP site (http://www.openldap.org/doc/admin/quickstart.html — start at "Edit the configuration file" since the LDAP files are already installed) or see the LDAP Linux HOWTO (http://www.redhat.com/mirrors/LDP/HOWTO/LDAP-HOWTO.html) for instructions on using LDAP on your system. Both of these documents cover the rest of these steps in more detail.
3. Edit the /etc/openldap/slapd.conf file to get it right for your system. (See Section 4.6.1, *Edit /etc/openldap/slapd.conf* for more information on editing slapd.conf.)
4. Start slapd by typing /etc/rc.d/init.d/ldap start. (After you have configured LDAP correctly, you should use Linuxconf or ntsysv to configure LDAP to start up with the system.)
5. Create your LDAP directory (examples of LDAP entries are provided at the PADL Software website at http://www.padl.com/ldap_examples.html).
6. Add entries to your LDAP directory with ldapadd or with a script.
7. Use ldapsearch to see if slapd is working.
8. At this point, your LDAP directory should exist. The next step is to configure your LDAP-enabled applications so that they can use the LDAP directory.

4.10 Configuring Your System to Authenticate Using OpenLDAP

This section provides a brief overview of how to configure your Red Hat Linux system to authenticate using OpenLDAP. Unless you're an OpenLDAP expert, you will probably need more documentation

than is provided here. Please refer to the references provided in Section 4.11, *Additional Resources* for more information.

4.10.1 Install the Necessary LDAP Packages

First, you should make sure that the appropriate packages are installed on both the LDAP server and the LDAP client machines. The LDAP server needs the `openldap` package.

The LDAP client machines need the following packages installed: `openldap`, `auth_ldap`, and `nss_ldap`.

4.10.2 Edit the Configuration Files

Edit `/etc/openldap/slapd.conf`

Next, edit the `slapd.conf` file to make sure it matches the specifics of your organization.

Please refer to Section 4.6.1, *Edit `/etc/openldap/slapd.conf`* for instructions on editing `slapd.conf`.

Edit `ldap.conf`

Edit the `ldap.conf` files in `/etc` and in `/etc/openldap` on the LDAP server and clients.

Edit `/etc/ldap.conf`, the configuration file for `nss_ldap` and `pam_ldap`, to reflect your organization and search base. The file `/etc/openldap/ldap.conf` is the configuration file for the command line tools like `ldapsearch`, `ldapadd`, etc., and it will also need to be edited for your LDAP setup. Client machines will need to have both of these files modified..

Edit `/etc/nsswitch.conf`

To use `nss_ldap`, you'll need to add `ldap` to the appropriate fields in `/etc/nsswitch.conf`. (Be very careful when editing this file; be sure that you know what you're doing.) For example:

```
passwd: files ldap
shadow: files ldap
group: files ldap
```

PAM and LDAP

To have standard PAM-enabled applications use LDAP for authentication, run `authconfig` and select **Use LDAP**. (PAM is beyond the scope of this LDAP overview, so if you need help, consult Chapter 8, *Pluggable Authentication Modules (PAM)* and the PAM man pages.)

4.10.3 Migrate Your Old Authentication Information to LDAP Format

The `/usr/share/openldap/migration` directory contains a set of shell and Perl scripts for migrating your old authentication information into LDAP format. (You must have Perl installed on your system to use these scripts.)

First, you'll need to modify the `migrate_common.ph` file so that it reflects your domain. The default DNS domain should be changed from:

```
$DEFAULT_MAIL_DOMAIN = "padl.com";
```

to something like:

```
$DEFAULT_MAIL_DOMAIN = "your_company.com";
```

The default base should also be changed, from:

```
$DEFAULT_BASE = "dc=padl,dc=com";
```

to something like:

```
$DEFAULT_BASE = "dc=your_company,dc=com";
```

Next, you'll need to decide which script to use. The following table may help you:

Table 4–1 LDAP Migration Scripts

Existing name service	Is LDAP running?	Use this script:
`/etc` flat files	yes	`migrate_all_online.sh`
`/etc` flat files	no	`migrate_all_offline.sh`
NetInfo	yes	`migrate_all_netinfo_online.sh`
NetInfo	no	`migrate_all_netinfo_offline.sh`
NIS (YP)	yes	`migrate_all_nis_online.sh`
NIS (YP)	no	`migrate_all_nis_offline.sh`

Run the appropriate script based on your existing name service.

The README and the migration-tools.txt files in /usr/share/openldap/migration provide more details on how to migrate the information.

4.11 Additional Resources

A lot of useful information concerning LDAP is available. Please review these sources, especially the OpenLDAP website and the LDAP HOWTO, before beginning to set up and configure LDAP on your system.

4.11.1 Installed Documentation

- The ldap man page is a good place to get started for an introduction to LDAP. Also, man pages exist for the various LDAP daemons and utilities. Please check the man pages if you need more information on ldapmodify, ldapsearch, and the like.
- /usr/share/docs/openldap-*versionnumber* — Contains a general README document and miscellaneous information.

4.11.2 Useful Websites

- http://www.openldap.org — Home of the OpenLDAP Project, the collaborative effort to develop a "robust, commercial-grade, fully featured, and open source LDAP suite of applications and development tools."
- http://www.redhat.com/mirrors/LDP/HOWTO/LDAP-HOWTO.html — LDAP Linux HOWTO document, covering the installation through authentication and logging.
- http://www.padl.com — Developers of nss_ldap and pam_ldap, among other useful LDAP tools.
- http://www.innosoft.com/ldapworld — Contains information concerning LDAP RFCs and LDAP version 3 specifications.
- http://www.kingsmountain.com/ldapRoadmap.shtml — Jeff Hodges' LDAP Road Map contains links to several useful FAQs and emerging news concerning the LDAP protocol.
- http://www.rudedog.org/auth_ldap — Home of the auth_ldap authentication module for Apache.
- http://www.stanford.edu/~bbense/Inst.html — Discusses the use of LDAP with Sendmail.
- http://www.webtechniques.com/archives/2000/05/wilcox — A useful look at managing groups in LDAP.
- http://www.ldapman.org/articles — Articles that offer a good introduction to LDAP, including methods to design an directory tree and customizing directory structures.

4.11.3 Related Books

- *Implementing LDAP* by Mark Wilcox; Wrox Press, Inc.
- *Understanding and Deploying LDAP Directory Services* by Tim Howes et al.; Macmillan Technical Publishing

5 Credit Card Verification System (CCVS) Basics

The Credit Card Verification System (CCVS) uses your computer and modem to simulate a credit card swipe box, also known as a **Point of Sale** (**POS**) terminal. A stand-alone product, CCVS includes several Application Programming Interfaces (APIs) that facilitate customization and integration with third party software applications or database products.

CCVS is safe, secure, and easy to use. Written in ANSI C and conforming to POSIX standards, CCVS is portable and easily integrated with modern operating systems, programming languages, and the Internet. Designed for easy scripting and programming, CCVS can be used to automate batch processing or enhance any application that requires credit card processing.

CCVS can be used in countries other than the US if your bank or merchant services representative uses one of the protocols supported by CCVS. If you are in Canada, CCVS supports the NDC protocol, which can be used by any bank in Canada to configure your merchant account. If you're in a country other than the US or Canada, you will need to check with your merchant services representative. The protocol supported by CCVS that has the best chance of being supported by a financial institution outside the US is the the Visa 2nd Generation "K Format" protocol (VITAL).

A demonstration version of CCVS is included with Red Hat Linux. The demo version is fully functional and can be used for testing CCVS on your system. In demo mode, it will do everything except connect to your financial institution and transmit the payment request. If you choose to use CCVS in a production environment to process credit cards, contact Red Hat to purchase a license key. See http://www.redhat.com/products/software/ecommerce/ccvs for more information on how to activate CCVS.

5.1 Uses for CCVS

CCVS excels at making the connection between an e-commerce application and a credit card payment gateway. While the ways in which you can use CCVS depend upon the protocol your payment gateway uses, in many cases, CCVS can be used with very few changes to an existing system. See http://www.redhat.com/products/software/ecommerce/ccvs/support/docs/protocol-specific.html for specific information on the different protocols supported by CCVS.

Consider the following examples of how CCVS can be used:

- CCVS can support a system for telephone operators taking catalog orders over the phone. The CCVS Tcl extensions can be used to create a Tcl/Tk Graphical User Interface (GUI) that presents a simple interface for telephone operators. The operators can then use simple X terminals, and all of the software will run on the central server. CCVS only needs to be installed on one computer,

and the operators don't have to wait for an available phone line — all of their transactions will go out over the same phone call.

- CCVS can be used to help automate billing. For example, an Internet Service Provider (ISP) might have a customer database on a database server. The ISP's database administrator could write a Perl script, combining the CCVS Perl module with a module for the ISP's database system. The script would then be run every month. The script will read the customer data, process monthly billing, and update the records in the database to indicate payment has taken place.
- CCVS can be used to help process payments for a web storefront which also uses a call center to handle telephone orders. In this way, orders processed over the web using a standard CGI application or by a sales agent using a custom Java program running over the LAN can go through the same connection for processing and payment. In addition, CCVS Address Verification System (AVS) features can be used to prevent fraud in both order methods without having to worry about implementing this feature separately in each of the applications, therefore saving development time.

These are only a few examples of the capabilities of CCVS. It can be used to enhance any aspect of your operations that require credit card processing. The many features of CCVS include the following:

- A C library with a documented API empowers users to integrate CCVS seamlessly with existing applications.
- A Tcl extension enables use of CCVS with server-side Tcl such as NeoWebScript.
- A Perl 5.0 module allows CCVS to work with the most popular CGI programming language in use today.
- The ability to quickly construct custom GUIs using Tcl/Tk — typical development time is less than a day.
- Python, PHP3 and Java modules allow CCVS to work with other common programming languages.
- Command Line Interface (CLI) programs for interactive use — call programs from any UNIX shell and program in the UNIX language you like best.
- AVS fraud protection, which allows merchants to check for stolen credit cards. Many clearinghouses offer a better rate to merchants who use AVS, even on orders taken over the phone.
- Support for multiple merchant accounts, allowing users to open their own virtual malls with unlimited store fronts. A **merchant account** is a special type of bank account which allows a business to accept credit card payments from customers. The merchant account holds the proceeds from credit card transactions.
- The ability to conduct multiple transactions in a single session, approaching leased line performance (two seconds per transaction!) with no extra cost or complexity.

- The reassurance of being able to test and do development programming on the product without charging real credit cards.

5.2 The Credit Card Verification Process

How does a little piece of plastic tell the salesperson that you can afford to buy that TV?

First, a consumer presents their credit card information to the merchant. The merchant transmits this data, along with their merchant ID code, to a clearinghouse (also referred to as a processor or acquirer). The clearinghouse might be the bank that has issued the merchant their credit card account, but it is more likely a firm that has contracted with the merchant's bank to clear charges in exchange for a flat fee and a percentage of every charge processed.

The data is transmitted by reading the card and merchant numbers over the phone, by using a credit card POS terminal, or by using CCVS or some other piece of software to transmit the information from a computer.

The clearinghouse contacts the bank that issued the consumer's credit card and verifies that the charge is acceptable. If it is accepted, the clearinghouse then sends a confirmation message to the merchant. At the same time, the available credit from the customer's credit card is frozen by the amount of the transaction.

At the end of a business day, the merchant (actually, the merchant's computer or credit card terminal) calls the clearinghouse and verifies all transactions for that day to ensure that the merchant's system and the clearinghouse agree on the transactions that have occurred during that day. Once the merchant and the clearinghouse agree on the day's transactions, the clearinghouse starts the process of transferring the money from the credit card bank to the merchant's bank account.

5.3 What You'll Need to Run CCVS

To run CCVS, you will need a modem and a merchant account. You should also follow a few guidelines so that CCVS will run correctly.

5.3.1 Modems

You need at least one modem dedicated to CCVS use. Credit card protocols do not support compression or error correction during modem connects, so compression and error correction cannot be used. Red Hat can provide you with information about how to turn off such features on the following modems:

- Hayes Optima
- US Robotics Courier
- US Robotics Sportster

- Chase Research PCI-RAS

Note

Please use a modem or modems from the above list!

If you use a non-supported modem (anything besides the four modems listed above), it may be very difficult to get the unsupported modem to work with CCVS. You should also check the Red Hat Linux Hardware Compatibility Lists at http://hardware.redhat.com to make sure that your modem will work with Red Hat Linux.

If the modem you must use does not appear on this list, look through your modem's manual to find the string which turns off *all* compression and error correction, as well as the string which resets your modem for normal use. You will need to provide these two strings when you configure CCVS.

5.3.2 Merchant Accounts

If you are just setting up a merchant account or modifying an existing merchant account in order to use CCVS, your merchant account provider may want to see proof that CCVS can work with the protocol it uses. Certification letters for specific protocols are available at http://www.redhat.com/products/software/ecommerce/ccvs/support/certifications.html. Print all pages of the letter corresponding to the protocol you will be using and show it to your merchant account provider.

Your merchant account provider must use one of the protocols supported by CCVS:

- First Data Corporation's ETC PLUS protocol (also known as FDR7, ETC+, ETC7, Omaha)
- First Data Corporation's South Platform protocol (also known as Nabanco)
- Global Payment Systems' MAPP protocol (also known as St. Louis)
- Global Payment Systems' NDC protocol (also known as Atlanta)
- Visa International's VITAL protocol (also known as VisaNet, Visa 2nd generation, K format)
- Paymentech's UTF protocol (also known as GENSAR)
- NOVA Information Systems protocol

If your merchant account provider one of these protocols, you will be able to use CCVS.

Once you've identified which protocol you will be using, review the information applicable to that protocol at http://www.redhat.com/products/software/ecommerce/ccvs/support/docs/protocol-specific.html before you start the CCVS configuration process. The *CCVS Protocol Guide* describes the functionalities supported by different protocols.

5.3.3 Guidelines for Using CCVS On Your System

The following requirements allow CCVS to run smoothly and efficiently. Please make sure you are following all these guidelines before attempting to run CCVS.

Exclusive Use of the Modem(s) While CCVS is Running

Do not run other software applications that need to access the modem while you are running CCVS. They can interfere with proper operation of CCVS by making the modem unavailable and preventing credit card numbers from being processed.

Permissions, Privileges and Access to the Modem

Most of the permissions needed for CCVS are set up for you during the installation process through the creation of a special group called ccvs. However, you will need to be aware of certain issues involving system permissions and CCVS. These issues are detailed in this section.

All operations for a particular CCVS configuration must be performed from a single user account. One account is required so that all file ownerships and permissions are correctly set and protected. This user account must be added to the ccvs group (by you or by your system administrator) before you run the configuration program.

After the user has been added to the ccvs group, run the CCVS configuration program (`ccvs_configure`) as that user. After you've run the configuration program, the same user must run the CCVS commands for that configuration.

If you want CCVS to run with a modem, the users in the ccvs group must also be added to the uucp group. Membership in the uucp group may not be sufficient for running the modems. If this is the situation on your system, be sure that the ccvs group members also have access to the serial port for the modem(s) that CCVS needs to use.

If you are using PHP with CCVS, enable the web server to run CCVS commands. To accomplish this, you must make the web server user a member of the ccvs group. Usually, the web server user will also need to be a member of the uucp group.

If you are not using PHP but want to make your web server capable of running CCVS, you have other options (such as `suexec` or `setuid`) other than making the web server user a member of the ccvs group. You can set it up any way you like, unless you're using PHP.

Software Versions

CCVS requires Tcl version 7.6 or greater to run the included GUI or to use the included Tcl/Tk APIs to develop your own graphical front end. Tcl version 8.3 is included in Red Hat Linux 7.1.

CCVS requires Perl version 5.0 or greater to use the included Perl APIs. Perl version 5.6 is included in Red Hat Linux 7.1.

5.4 Installing CCVS

The CCVS RPMs are available on the Linux Applications Library Workstation CD.

You can use RPM, Gnome-RPM or Kpackage to install the CCVS packages:

- `CCVS` — The core CCVS programs
- `CCVS-devel` — The C developer's kit
- `CCVS-perl` — The Perl interface for CCVS
- `CCVS-python` — The Python interface for CCVS
- `CCVS-php3` — The PHP3 interface for CCVS
- `CCVS-tcl` — The Tcl interface for CCVS
- `CCVS-java` — The Java interface for CCVS (included as source code)
- `CCVS-examples` — Sample source code, needed for development

5.5 Before You Configure CCVS

Before configuring CCVS, you need to be able to answer certain questions about your system and about how you want to set up CCVS. To prepare for the configuration process, be sure to follow these steps:

1. Please read through all documentation and errata that came with the program. See Section 5.11, *Additional Resources* for the locations of installed and online CCVS documentation.
2. Fill out `setup.txt`, which is a worksheet file explaining the different pieces of information needed when configuring CCVS to use particular protocols. If you fill out `setup.txt`, you will have all of the information required for the configuration process available at your fingertips, preventing any surprises when you run the configuration program. You can find `setup.txt` in the `/usr/share/doc/CCVS-<version>` directory. Alternatively, it is also available at http://www.redhat.com/products/software/ecommerce/ccvs/support/docs/setup.txt.

Note

On the setup worksheet, you'll be asked for some protocol-specific information. You only need to provide information for the protocol which you are going to use. You don't need to fill in the worksheet information for any of the other protocols.

---

3. The CCVS configuration program will ask you several things about your modem, so be prepared with the appropriate information. The following init strings can be used with the supported modems:

Hayes Optima or ACCURA

```
\r~~~\rAT &D3 X4 E0 &K0 &Q0
```

U.S. Robotics Sportster or Courier

```
\r~~~\rAT E0 L0 M1 V1 X4 &K0 &M0 +FCLASS=0
```

Chase Research PCI-RAS

```
\r~~~\rAT E0 %C0 \\N0
```

If using one of the supported modems, the configuration program will only ask you to confirm the init string. If your modem is not listed above, look through the modem's manual to find the string which turns off all compression and error correction, as well as the string which resets your modem for normal usage. You will need to set these two modem strings during the configuration process.

5.6 Configuring CCVS

You must configure CCVS for your system, whether you're running CCVS in demo mode or for processing real data.

Use `su` to switch to the user account that you created (a member of the ccvs group) for this configuration.

Run the CCVS configuration program with the following command:

```
/usr/sbin/ccvs_configure
```

The rest of this section will walk you through the CCVS configuration program. You should see an initial "splash" screen. Press [Enter] to read the CCVS software license. You can use the standard scrolling and paging commands of `more` (or the paging program set by your $PAGER environment variable) to read the license.

When you have read the license and exited the pager, you will see:

```
Type "accept" to accept this license, or anything else to exit.
```

Type the word **`accept`** to accept the terms of the license and continue configuring CCVS. Any other input will exit the program.

You will then see this screen:

```
This program creates the configuration file for CCVS functions.
```

```
To do this, you will require the following information:
 1: The clearing protocol you will be using. This may be MAPP,
ETC+, or any of the other protocols which CCVS supports. There
is also a demo protocol; if you have downloaded the free demo of
CCVS, you will be using the demo protocol.
 2: The unique number which identifies you to the clearing
house. This may be your merchant account number or a terminal id
number, depending on what protocol you will be using. This number
will be supplied when you set up your merchant account.
 3: Your modem type, and the serial port your modem is attached
to. You will also need modem configuration strings. (We can
supply modem configuration strings for many popular modems.)
 4: The location of your data directory. This is where the
configuration file and data directories will be placed.
 5: Other information as needed for particular protocols. This
information will generally be supplied when you set up your
merchant account.

We supply a worksheet which you can use to organize all this
information, including the details for each protocol. See the
file "setup.txt" in /usr/share/doc/CCVS-<version>.

The configuration program is running as user "<username>".
It is important that this be the same user which the actual CCVS
software will run as. (We recommend creating a special user
account for just this purpose.)

Do you wish to continue configuring CCVS as user "<username>"?

[Enter Y to continue, or N to stop here:]
```

Press [Y] to continue. If you are running `ccvs_configure` as root, you will instead get the following error. If this happens, you should `su` to the CCVS user, such as the default ccvs user, and re-run `ccvs_configure`.

```
The configuration program may not be run as root. You must run
this as the same user which the actual CCVS software will run as.
(We recommend creating a special user account for just this
purpose.)
```

When you continue, the program will begin prompting you for information. At any time, you can back up to a previous prompt by typing **.** (a period) by itself and pressing [Enter].

```
Do you want to configure CCVS for the free demo, or a working
merchant account? (If you have not purchased a license for CCVS,
only the demo configuration is available.)
```

```
[Enter Y to use the demo configuration, N for a real configuration,
or . to exit:]
```

Unless you have purchased a software key and license for CCVS, type [Y]. This installs a demo configuration, which does not dial the modem or use a real merchant account. If you have purchased a license and are ready to install a working configuration, type [N].

```
Where do you want to place the CCVS configuration files and
transaction queues? This should be a directory name which is
writable by the current user.
The default is "/var/ccvs".
Enter directory, or Return for default value, or . by itself to
back up.
>
```

Unless you have specific reasons for moving the CCVS configuration files and transaction queues, leave them in their default locations. If you need to move them, remember that you will also need to set an environment variable.

```
What do you want to name this configuration? This should be a
short filename.
The default is "ccvs".
Enter name, or Return for default value, or . by itself to back
up.
>
```

For example, you might have a configuration called **`tshirt`** for a merchant who sells T-shirts and **`music`** for the sheet music retailer. The name entered here is the name used to distinguish between the two configurations.

The demo version of CCVS requires no other information; if you chose it, you will immediately see:

```
Writing "/var/ccvs/ccvs.conf"...

The CCVS system is now configured.
```

You can now begin testing the demo software. The demo acts just like the full CCVS software, except that it does not dial the modem or talk to a real merchant processor.

If you have a license for the full version of CCVS, and you chose to install a real configuration, you will instead see something like this:

```
Which protocol and merchant processor will you be using?

Credit card clearing protocols:
  1: ETC PLUS (FDR7/ETC7/FDR "Omaha"): First Data Corporation
  2: South Platform (FDR "Nabanco"): First Data Corporation
```

```
  3: MAPP: Global Payment Systems "St. Louis"
  4: NDC: Global Payment Systems "Atlanta" / NDC
  5: VITAL (Visa 2nd generation, K format): Visa/Total System Services
  6: UTF: Paymentech Inc.
  7: NOVA: NOVA Information Systems

[Enter a number, or . by itself to back up:]
```

Select the protocol for which you have a CCVS license and a valid merchant account.

```
What is the number of your merchant account?
Enter number, or . by itself to back up.
>
```

This number should have been provided with your merchant account.

```
What is your CCVS software customer number?
Enter number, or . by itself to back up.
>
```

This number will have been provided with your CCVS license.

```
What is your CCVS software license key?
Enter number, or . by itself to back up.
>
```

This number will also have been provided with your CCVS license.

```
What is the phone number of your merchant processor?
Enter number, or . by itself to back up.
>
```

Additional questions may appear, because different information is required by particular protocols. If you've filled in the `setup.txt` worksheet section for your protocol, you should be prepared for these questions. For example, VITAL continues with several more prompts about your business's name, address, bank, and so on. You should already have found out this information when you established your VITAL merchant account. This is the purpose of the `setup.txt` worksheet file, which you should have completed prior to running the CCVS configuration program. See Section 5.5, *Before You Configure CCVS* for information concerning the use of `setup.txt`.

You must now enter information about how to communicate with your modem. The modem configuration information is very important. Be sure that you enter correct information for your system's setup. CCVS will not work if the modem is set up incorrectly.

```
Do you want to configure a pool of several modems? (If you answer
yes, all the modems must be exactly the same make and model. If
you want to use just one modem, answer no.)
```

```
[Enter Y or N, or . to back up:]
```

If you have several identical modems, you can configure CCVS to use them all, as a pool. Each CCVS process which needs to use a modem will draw one from the pool, assuming one is available. Several CCVS configurations can share a collection of modems this way. You can also configure a single configuration with two modems, so that authorizations and batch settlement can occur at the same time.

```
What serial port is your modem connected to? (Do not include the
"/dev/" prefix.) The default is ttyS0. The modem should be
connected and ready now, so that the serial port can be tested.

Enter port name, or Return for default value, or . by itself to
back up.
>
```

The program will test the serial port you enter. If you configure more than one, it will test each of them. Don't include the /dev/. This step may take up to thirty seconds if the modem does not respond.

```
What type of modem do you have? This information makes it
possible to suggest modem configuration strings. If your modem
is not listed, you can choose "none of the above"; but you will
then have to create your own configuration strings, which is a
difficult process.

 1: USR Sportster/Courier
 2: Hayes Optima
 3: Chase Research PCI-RAS
 4: None of the above

[Enter a number, or . by itself to back up:]
```

You will be prompted for the modem initialization, dialing, and hang-up strings. (If you configure a pool of modems, they must all be identical, so they will all use the same strings.) If CCVS knows appropriate strings for your modem, they will be suggested and you can just press [Enter].

```
The modem initialization string should set the modem to do no
protocol
negotiation. What string do you want to use?
A string which works for your modem is:
 \r~~~\rAT E0 L0 M1 V1 X4 &K0 &M0 +FCLASS=0
Enter string, or Return for suggested value.
>

The modem dial string should dial the modem. (Do not include a
phone number.)
```

```
What string do you want to use?
A string which works for your modem is:
 ATDT
Enter string, or Return for suggested value.
>

The modem hang-up string should hang the modem up if it's
connected. What string do you want to use?
A string which works for your modem is:
 ~~~+++~~~~~~~~~\rATH0\r~~~
Enter string, or Return for suggested value.
>

Initialize: \r~~~\rAT E0 L0 M1 V1 X4 &K0 &M0 +FCLASS=0
Dial: ATDT
Hang up: ~~~+++~~~~~~~~~\rATH0\r~~~
Are these the values you want?

[Enter Y to accept, N to change, . to back up.]
```

You may not see exactly the same screen as shown above because the suggested defaults will vary depending on the modem you selected.

The next question concerns the baud rate the modem will use:

```
What baud rate do you want to use? You should use the
default unless you have explicit information that another
value is appropriate.
The default baud rate is 1200.

Enter rate, or Return for default value, or . by itself to
back up.
>
```

When you have finished entering configuration information, you will see:

```
Writing "/var/ccvs/ccvs.conf"...

The CCVS system is now configured.
```

5.7 Multiple Merchant Accounts

If you need to support more merchant accounts, simply follow the configuration procedure again. Use a different configuration name for each merchant account.

Different configurations may share the same serial port or the same pool of serial ports. The modems will be used first-come-first-served.

5.8 Starting CCVS

To run CCVS for a particular application, you should be logged in as the user account who created that configuration. If you are using a ccvs user account for this purpose and are already logged into the system as a different user, type `su ccvs` to switch to the correct user.

When logged in as the user set up to run CCVS programs, you will need to start the `ccvsd` daemon for each merchant account. In addition, you will need to run the `cvupload` program on a regular basis. Using `cron` to run `cvupload` every day will accomplish this task; see the `cron` man pages for instructions concerning automating processes.

5.8.1 The ccvsd Daemon

To run CCVS, you must run the `ccvsd` daemon. The `ccvsd` daemon actually makes the phone calls and conducts the transactions. The `ccvsd` command must be followed by the name of the account specified when you configured the account.

For example, if you want to start processing transactions for the example sheet music retailer mentioned during the configuration program, and you installed the software in its default location of `/usr/sbin`, you would type in the following command to start `ccvsd`:

```
/usr/sbin/ccvsd music
```

Every time you add a merchant account, you need to start `ccvsd` for that account, if you want to process transactions for that account.

For more information on `ccvsd`, see the `ccvsd` man page.

5.8.2 The cvupload Command

Some transactions (such as authorizations) occur at the time that the credit card is presented. Other transactions (such as sales and returns) are saved up and are not processed immediately. These transactions are gathered together and are then processed as a group.

CCVS uses the `cvupload` program to do this batch processing. We recommend invoking `cvupload` as an (at least) daily `cron` job, so that `cvupload` will automatically run every day, without any intervention on your part.

For example, the following command would be used to do the periodic processing for the sheet music retailer:

```
/usr/sbin/cvupload music
```

For more information on `cvupload`, see the `cvupload` man page.

5.9 Special Language Considerations

- C — The CCVS C library is included in the `CCVS-devel` package. When compiling C programs that use CCVS, add the `-lccvs` flag on the linkage line.
- Java — Please see http://www.redhat.com/products/software/ecommerce/ccvs/support/docs/AdminJava.html for more information on building the CCVS Java interface. The source code for the Java interface is provided in the `CCVS-java` package.
- Perl — The Perl interface is provided in the `CCVS-perl` package.
- Python — The Python interface is provided in the `CCVS-python` package.
- PHP — The PHP3 interface is provided in the `CCVS-php3` package.
- Tcl — The Tcl interface is included in the `CCVS-tcl` package.

5.10 Support for CCVS

Support for CCVS can be purchased from Red Hat. When you purchase your key to activate CCVS, be sure to review the support options available. See http://www.redhat.com/products/software/ecommerce/ccvs for more information about purchasing a key and obtaining support for CCVS.

If you do need support, be sure to have the following information available before you contact support:

- Your company name
- The version of CCVS you are using
- Your merchant number
- Your CCVS customer number
- Your operating system and version

Red Hat technical support will attempt to address any issues that deal directly with CCVS. We cannot support third party products, except for issues regarding integration with CCVS.

5.11 Additional Resources

Additional information concerning CCVS is available.

5.11.1 Installed Documentation

- `/usr/share/doc/CCVS–<version-number>` — Contains the `CHANGES`, `LICENSE`, and `README` files, plus the `setup.txt` worksheet to help gather the necessary information before running the configuration program.

- Type `man ccvs` for a description of the different states of a transaction, CCVS error codes, and much more. The man pages for `ccvsd`, `cvreport`, and `cvupload` describe a number of different options that can be used with those commands.

5.11.2 Useful Websites

- http://www.redhat.com/products/software/ecommerce/ccvs — From this location, you can link to many different CCVS resources, including FAQs, technical specifications, and general information about CCVS.
- http://www.redhat.com/products/software/ecommerce/ccvs/support/documentation.html — Contains links to several guides, written specifically to explain how to use CCVS in different ways. These online manuals cover everything from installing and configuring CCVS to a full description of the APIs for the various languages that can be used.

6 Sendmail

6.1 Introduction to Sendmail

Sendmail is a very popular **mail transfer agent** (**MTA**) used on the Internet, handling a very large percentage of all Internet-routed email at some point as it moves from one host to the other. Other mail transfer agents do exist (and can be used well with Red Hat Linux), but most administrators elect to use Sendmail as their MTA due to its power, scalability and compliance to Internet standards.

Sendmail's core duty, like other MTAs, is to safely move email between hosts, usually utilizing the **Simple Mail Transfer Protocol** (**SMTP**). However, Sendmail is highly configurable, allowing you to control almost every aspect of how email is handled.

Sendmail's roots can be traced to the birth of email, occurring in the decade before the birth of ARPANET, the precursor to the Internet. In those days, every user's mailbox was a file that only they had rights to read, and mail applications simply added text to that file. Every user had to wade through their mail file to find any old mail, and reading new mail was a chore. The first actual transfer of a mail message file from one host to another didn't take place until 1972, where email began to to be moved by FTP over the NCP network protocol. This easier method of communication quickly became popular, even to the point where it made up most of ARPANET's traffic in less than a year. However, a lack of standardization between competing protocols made email much harder to send from some systems, and this continued until the ARPANET standardized on TCP/IP in 1982. A new protocol, SMTP, materialized for message transporting. These developments, combined with HOSTS files being replaced with DNS, allowed full-featured MTAs to materialize. Sendmail, which grew out of an earlier email delivery system called Delivermail, quickly became the standard as the Internet began to expand and be widely utilized.

It is important to be aware of what Sendmail is and what it can do for you as opposed to what it is not. In these days of monolithic applications that fulfill multiple roles, you might initially think that Sendmail is the only application you need to run an email server within your organization. Technically, that is true, as Sendmail can spool mail to your users' directories and accepts new email via the command line. But, today's users actually require much more than simple email delivery. They almost always desire to interact with their email using a **mail user agent** (**MUA**) that utilizes the **Post Office Protocol** (**POP**), **Internet Message Access Protocol** (**IMAP**), or even the Web. These other protocols can work in conjunction with Sendmail and SMTP, but they actually exist for different reasons and can operate separately from one another.

It is beyond the scope of this chapter to go into all that Sendmail should or could be configured to do. Rather, consult the many excellent online and offline sources of information on Sendmail in order to shape it to fit your exact specifications. You should, however, understand what files are installed with Sendmail by default on your system, know how to make basic configuration changes, be aware of

how to stop unwanted email (spam) being sent through Sendmail, and know how to extend Sendmail with the **Lightweight Directory Access Protocol** (**LDAP**).

6.2 The Default Sendmail Installation

While you can download the source code for Sendmail and build your own copy, many users prefer to install Sendmail via RPM from the CD-ROM (at the time of the Red Hat Linux installation or at a later point).

The Sendmail application is placed in `/usr/sbin`.

Sendmail's lengthy and detailed configuration file (`sendmail.cf`) is installed in `/etc`. You should not edit the `sendmail.cf` file directly unless you know exactly what you are doing, due to the fact it is very lengthy and complex. Instead, to make configuration changes to Sendmail, edit the `/etc/mail/sendmail.cf` file and use the included `m4` macro processor to create a new `/etc/sendmail.cf` (after backing up the original `/etc/sendmail.cf`, of course). More information on configuring Sendmail can be found in Section 6.3, *Common Configuration Changes*.

Various Sendmail configuration files are installed in `/etc/mail` including:

- `access` — Specifies which systems can use Sendmail for relaying email.
- `domaintable` — Allows you to provide domain name mapping.
- `local-host-names` — The place where you include all aliases for your machine.
- `mailertable` — Specifies instructions that override routing for particular domains.
- `virtusertable` — Permits you to do a domain-specific form of aliasing, allowing multiple virtual domains to be hosted on one machine.

Several of the configuration files in `/etc/mail`, such as `access`, `domaintable`, `mailertable` and `virtusertable`, must actually store their information in database files before Sendmail can use any configuration changes. To include any changes you make to these configuration in their database files, you must run a command with the syntax `makemap hash /etc/mail/`*`name`* `< /etc/mail/`*`name`* where *name* is the name of the configuration file to convert.

For example, if you want all email addressed to any `domain.com` account to be delivered to `bob@otherdomain.com`, you need to add a line to the `virtusertable` file:

```
@domain.com        bob@otherdomain.com
```

Then, to add this new information to the `virtusertable.db` file, execute `makemap hash /etc/mail/virtusertable < /etc/mail/virtusertable` as root. This will create a new `virtusertable.db` that contains the new configuration.

6.3 Common Configuration Changes

A default `sendmail.cf` file will be installed in `/etc`. The default configuration should work for most SMTP-only sites. It will *not* work for UUCP (UNIX to UNIX Copy) sites; you will need to generate a new `sendmail.cf` if you must use UUCP mail transfers.

Note

Although SMTP servers are supported automatically, **IMAP** (Internet Message Access Protocol) servers are not. If your ISP uses an IMAP server rather than an SMTP sever, you must install the IMAP package. Without it, your system won't know how to pass information to the IMAP server or retrieve your mail.

If you need to generate a new `/etc/sendmail.cf` file to configure Sendmail, you should utilize the `m4` macro processor. If you ever edit the `/etc/mail/sendmail.mc` to add functionality to Sendmail, backup your current `/etc/sendmail.cf` file, generate a new one by executing the `m4 /etc/mail/sendmail.mc > /etc/sendmail.cf` command, and add any previous changes from the `/etc/sendmail.cf` you backed up to the new `/etc/sendmail.cf` file. After creating a new `/etc/sendmail.cf`, you must restart Sendmail to make it take effect. The easiest way to do this is to type the `/sbin/service sendmail restart` command as root.

By default, the `m4` macro processor is installed with Sendmail. The `m4` macro processor is included with the `sendmail-cf` package, which is installed in `/usr/lib/sendmail-cf`.

You should consult the `/usr/lib/sendmail-cf/README` file before you edit any of the files in the directories under the `/usr/lib/sendmail-cf` directory, as they can affect how future `/etc/sendmail.cf` files are configured.

WARNING

Do not use Linuxconf to configure Sendmail! The Linuxconf module `mailconf`, designed to make editing `/etc/sendmail.cf` easier, is broken and contains out-of-date information about rule sets used in Sendmail configuration.

One common Sendmail configuration is to have a single machine act as a mail gateway for all the machines on your network. For instance, a company may want to have a machine called `mail.bigcorp.com` that does all our mail. On that machine, we simply need to add the names of machines

for which `mail.bigcorp.com` will handle mail to `/etc/mail/local-host-names`. Here is an example:

```
# sendmail.cw - include all aliases for your machine
# here.
torgo.bigcorp.com
poodle.bigcorp.com
devel.bigcorp.com
```

Then on the other machines, `torgo`, `poodle`, and `devel`, we need to edit `/etc/sendmail.cf` to "masquerade" as `mail.bigcorp.com` when sending mail and to forward any local mail processing to `bigcorp.com`. Find the `DH` and `DM` lines in `/etc/sendmail.cf` and edit them as such:

```
# who I send unqualified names to
# (null means deliver locally)
DRmail.bigcorp.com

# who gets all local email traffic
DHmail.bigcorp.com

# who I masquerade as (null for no masquerading)
DMbigcorp.com
```

With this type of configuration, all mail sent will appear as if it were sent from `bigcorp.com`, and any mail sent to `torgo.bigcorp.com` or the other hosts will be delivered to `mail.bigcorp.com`.

Please be aware that if you configure your system to masquerade as another, any email sent from your system to your system will be sent to the machine you are masquerading as. For example, in the above illustration, log files that are periodically sent to `root@poodle.bigcorp.com` by the `cron` daemon would be sent to `root@mail.bigcorp.com`.

6.4 Stopping Spam

Email **spam** can be defined as unnecessary and unwanted email received by a user that probably doesn't know the sender and never requested the communication. It is a very disruptive, costly, and widespread abuse of Internet communication standards.

Thankfully, Sendmail has made it (relatively) easy to block new spamming techniques being employed to send junk email. It even blocks many of the more usual spamming methods by default, so that you would need to consciously activate them by changing your `/etc/mail/sendmail.cf` file in a particular way to make your system susceptible. For example, forwarding of SMTP messages, also referred to as **SMTP relaying**, has been disabled by default since the 8.9 version of Sendmail. Before this change occurred, Sendmail would direct your mail host (`x.org`) to accept messages from one party (`y.com`) and send them to a different party (`z.net`). Now, however, you have

to specifically tell Sendmail to permit a domain to relay mail through your domain. Simply edit `/etc/mail/relay-domains` and restart Sendmail by typing the `/sbin/service sendmail restart` command as root to activate the changes.

However, many times, your users may be bombarded by spam from other servers throughout the Internet beyond your control. In these instances, you can use Sendmail's access control features available through the `/etc/mail/access` file. As root, simply add the domains that you would like to block or specifically allow access, such as:

```
badspammer.com        550 Go away and don't spam us anymore
tux.badspammer.com    OK
10.0                  RELAY
```

Because `/etc/mail/access` is a database, you need to use makemap to activate your changes by recreating the database map. This is easily done by running the `makemap hash /etc/mail/access < /etc/mail/access` command as root.

This example shows that any email sent from `badspammer.com` to you would be blocked with 550 RFC 821 compliant error code and message back to the spammer, except for email send from the `tux.badspammer.com` sub-domain, which would be accepted. The last line shows that any email sent from the 10.0.*.* network can be relayed through your mail server.

As you might expect, this example only scratches the surface of what Sendmail can do in terms of allowing or blocking access. See the `/usr/share/doc/sendmail/README.cf` for more detailed information and examples.

6.5 Using Sendmail with LDAP

As we have already seen in Chapter 4, *Lightweight Directory Access Protocol (LDAP)*, Lightweight Directory Access Protocol (LDAP) is a very quick and powerful way to find specific information about a particular user from a much larger group. For example, you could use an LDAP server to look up a particular email address from a common corporate directory by a user's last name. In this kind of implementation, LDAP is largely separate from Sendmail, with LDAP storing the hierarchical user information and Sendmail only being given the result of LDAP queries in pre-addressed email messages.

However, Sendmail supports a much greater integration with LDAP, where it uses LDAP to replace separately maintained files, such as `aliases` and `virtusertables`, on different mail servers that work together to support a medium- to enterprise-level organization. In short, you can use LDAP to abstract the mail routing level from Sendmail and its separate configuration files to a powerful LDAP cluster that is being leveraged by many different applications.

The current version of Sendmail contains support for LDAP. To extend your Sendmail server using LDAP, first get an LDAP server, such as OpenLDAP, running and properly configured. Then, you need to edit your `/etc/mail/sendmail.mc` to include:

```
LDAPROUTE_DOMAIN('yourdomain.com')dnl
FEATURE('ldap_routing')dnl
```

Note

This is only for a very basic configuration of Sendmail with LDAP. Your configuration should differ greatly from this depending on your implementation of LDAP, especially if you wish to configure several Sendmail machines to use a common LDAP server.

Consult `/usr/share/doc/sendmail/README.cf` for detailed LDAP routing configuration instructions and examples.

Next, recreate your `/etc/sendmail.cf` file by running `m4` and restarting Sendmail. See Section 6.3, *Common Configuration Changes* for instructions on doing this.

For more information on LDAP, see Chapter 4, *Lightweight Directory Access Protocol (LDAP)*.

6.6 Additional Resources

Many users initially find Sendmail difficult to configure, primarily due to the large number of options available. Access to additional Sendmail documentation can be very helpful, especially when setting configuration options.

6.6.1 Installed Documentation

The best sources of information about how to configure Sendmail are included with the `sendmail` and `sendmail-cf` packagess.

- `/usr/share/doc/sendmail/README.cf` — Contains information on `m4`, file locations for Sendmail, supported mailers, how to access enhanced features, and much more.
- `/usr/share/doc/sendmail/README` — Contains information on the Sendmail directory structure, IDENT protocol support, details on directory permissions and the common problems these permissions can cause if misconfigured.

6.6.2 Useful Websites

- http://www.sendmail.net — News, interviews and articles concerning Sendmail, offering a larger view of the many options available.
- http://www.sendmail.org — Offers a thorough technical breakdown of Sendmail features and configuration examples.

6.6.3 Related Books

- *Sendmail* by Bryan Costales with Eric Allman et al; O'Reilly & Associates — A good Sendmail reference written with the assistance of the original creator of Delivermail and Sendmail.

Part II Security-Related Reference

7 Red Hat Security Primer

Beyond the proper installation and configuration of your Red Hat Linux system, it is critical that you secure the system to an acceptable level of risk given its role, importance, and expected use. Security is an incredibly complex subject that constantly involves emerging problems, as well as potential ones.

Due to its amorphous and intricate nature, many system administrators and users make the mistake of tackling small, isolated problems while letting much larger and dangerous issues slip by. True system security goes far beyond the installation of the latest update, the configuration of a certain file, or the careful administration of user access to system resources. It is a way of looking at the various threats to your system and the lengths you will go to prevent them.

No system is completely secure unless it is turned off (and even then, it is susceptible to being stolen). Any time the system is on, it is susceptible to attack, ranging from a harmless prank to a hardware-destroying virus to data being erased. But all is not lost. With the proper outlook, as well as some good tools, you can enjoy many years without experiencing a single security problem. The following sections are designed to outline a way to approach system security and potential threats, a context within which to consider various security tools, costs, and benefits when running Red Hat Linux.

7.1 The Inescapable Security Dilemma

All users of any operating system face a common dilemma when constructing a security paradigm for their system. On one hand, they seek to avoid making the system so secure that nothing will run on it properly. But on the other hand, they also try to avoid making the system so insecure that anyone can (and will) do anything on it they they wish to, including deleting the work of others or much worse scenarios.

There is no one right way to solve this dilemma. Some systems, either by the nature of their purpose or the importance of the data they protect, fall on one side of the dilemma while other systems, whether because of the wide variety of users utilizing them or the fact that they are test machines, fall on the other side.

The most important thing you can do when configuring the security of your system is to determine where on the security dilemma spectrum your particular system lies. This may be done for you by company policy. Or, you may be a researcher with a system that you never connect to public networks, and no one other than you has physical access to the machine. Or, you may be a home user that is connected to a broadband connection and (rightfully) concerned about ways malicious users a world away could damage your data.

Regardless which of the countless possible scenarios you may fit in, you bear the responsibility to determine your proper exposure to risk versus the goals your system must accomplish. Then, once

you make this determination, use this knowledge as a guide of how to set up and maintain security guidelines on your system.

7.2 Active vs. Passive Approaches

Security approaches can always be broken down into two different types: **active** or **passive**. An **active** approach to security covers all actions designed to prevent a breech of your system's security model. A **passive** approach to security refers to the actions taken to monitor the security of your system based on that security model.

All users should employ both active and passive approaches to security. Each of these approaches strengthens the other. The fact that you know from server logs that a particular user is trying to crack your security (passive approach to security) may lead to you install an application to block them from even getting a login prompt in the first place (active approach to security). Likewise, the fact that you are not using shadow passwords to protect your system (active) may lead you to watch vigorously for changes to key files on your system using a tool such as Tripwire (passive). (For more information on Tripwire, please see Chapter 10, *Installing and Configuring Tripwire*.)

Red Hat Linux includes a variety of tools that will help you implement both approaches to security. But the proper use of methods with each approach is crucial to prevent an over-dependence on tools to protect your system.

7.2.1 Tools and Methods for an Active Approach to Security

The vast majority of security tools for Red Hat Linux work to actively protect your system. Here are a few of the most common and useful open source tools:

- *Shadow Utilities* — A collection of industry-standard tools to administer local users and groups on a system using encrypted passwords.
- *Kerberos 5* — A secure system for providing network authentication services. Prevent the use of plaintext passwords being passed over a network to gain access to services. (See Chapter 9, *Using Kerberos 5 on Red Hat Linux* for more information on Kerberos 5.)
- *OpenSSL* — Helps you to protect a wide variety of services that support operation over an encryption layer. (See the *Official Red Hat Linux Customization Guide* for more information on OpenSSL.)
- *OpenSSH* — A set of utilities that can easily replace such ubiquitous yet insecure tools as `telnet` and `ftp` with the powerful and secure `ssh` and `scp`. (See the *Official Red Hat Linux Customization Guide* for more information on OpenSSH.)

Methods that support an active approach to security include the following:

- *Limiting the number of users that can execute commands as root* — Whether intentional or by accident, a large percentage of security problems result at least indirectly from someone knowing the root password or being given permission via `sudo` to perform a root-level command.
- *Knowing what software packages you have installed on your system and remaining alert for newly discovered security holes* — You won't know what packages to look out for unless you are aware of which ones are installed on your system, and you won't know they need updating unless you monitor sources of information, such as the Red Hat Network.
- *Limiting the services running on the system to only those that you actually need* — Basically, the more you have running, the more that can break or provide unauthorized access. Save system resources (and the trouble of maintaining things you don't use) and remove packages you aren't using. At the very least, run a tool such as `ntsysv` to prevent unnecessary services from starting with the system at boot. (See *Controlling Access to Services* in the *Official Red Hat Linux Customization Guide*.)
- *Require users to create secure passwords and change them often* — Most security problems begin with unauthorized access to the system. This risk can be minimized by requiring your users to also practice active security methods by protecting their keys to your gate.
- *Making sure file permissions aren't unnecessarily open* — Almost no files should be writable by all.

7.2.2 Tools and Methods for an Passive Approach to Security

While most security tools for Red Hat Linux are designed for an active approach to security, there are a few tools that can make passive security much less of an administrative burden:

- *Tripwire* — An application designed to alert you if specified system files and directories are changed. In this way, you will at least know if unauthorized users are gaining access to your system or authorized users are making unwanted changes to important files. (See Chapter 10, *Installing and Configuring Tripwire* for more information on Tripwire.)
- *COPS* — A collection of security tools designed to do a number of different things, from checking open ports on a particular host to looking out for poor user passwords.

Methods that support an passive approach to security include the following:

- *Making it a routine practice to monitor system logs* — By default, Red Hat Linux traps an enormous amount of useful data in the system logs located in the `/var/log` directory, especially in the `messages` file. One simple task run as root, such as the `grep "session opened for user root" /var/log/messages | less` command, allows you to perform a powerful partial audit on your system and monitor who is accessing the system as root. This would allow you, for example, to quickly narrow the number of possible users that could have changed a particular file that can only be written to by root, simply by comparing the time the file in question

was changed with the time of the logins in the `/var/log/messages` file. However, consider that this is not a foolproof method, as someone with write control over an important system file may also have rights to change `/var/log/messages` to erase their tracks.

7.3 Developing Security Policies

Every system, from a machine used by only one person to an enterprise-level server utilized by thousands of users, should have a security policy. A security policy is a set of guidelines used to gauge whether a particular activity or application should or should not be done or utilized on a system, based on the particular objectives for that system.

Security policies between different systems can vary greatly, but the most important thing is that one actually does exist for your system - whether or not is written down in company policy manual or simply remembered.

Any security policy should be constructed using these features as guides:

- *Simple rather than complex* — The more simple and straightforward the security policy, the more likely the guidelines will be followed and the system kept secure.
- *Easy to maintain rather than difficult* — Security methods and tools, like everything, are subject to change based on new challenges and needs. Your security policy should be built around minimizing the impact changes will have on your system and its users.
- *Promote freedom through confidence in the system's integrity rather than stifling system usability* — Avoid security methods and tools that unnecessarily decrease the usefulness of your system when making the system more secure. Quality security methods and tools are almost always win-win, making the system more secure while offering more choice to users wherever possible.
- *Recognition of fallibility rather than a false sense of security* — One of the most successful ways to invite a security problem is through the belief that your system could not possibly have that problem. Rather than resting on your laurels, be eternally vigilant.
- *Focus on real problems rather than being overoccupied by theoretical ones* — Spend your time and effort dealing with the biggest real problems and work down from there. Prioritize your efforts and plug the biggest holes first. To help determine what you should be on the lookout for first, consider turning to http://www.sans.org/topten.htm or similar web sites that detail precise security problems that really do post a threat and exactly what you can do about them.
- *Immediacy rather than procrastination* — Fix problems when you find them and determine that they pose a risk. Don't fall prey to thinking that you can take care of this at a later time. There really is no time like the present, particularly when your system is at stake.

If you find that your security policy is so restrictive that it prevents the system from being used in the way intended, then consider sufficiently changing the policy to loosen access to the system. In the same way, if you find that your system's security is continually being compromised, you should

change aspects of your security policy to restrict access. Most importantly, remember that a security policy is not a static document or idea. It must be amended as the needs of your system's objectives and users change. Continuously reconsider your current security policy in the reflection of real world requirements.

7.4 Beyond Protecting Root

Many users put most of their security emphasis in restricting the number of users that can gain root access on their system. While this is obviously a very good and an important first step, much more must be done to make a system secure. For one thing, security is only one part of the larger issue of system stability. Security issues are often intertwined with larger points of stability, and a successful system balances methods and tools used for security protection with an awareness of alternate ways in which similar damage can be inflicted.

First of all, if your system is used by multiple users and those users change, be sure to delete the accounts of old users immediately after those accounts are no longer being used. Better still, develop a clear and concise checklist of items that must be done when a user account or group is no longer required.

Limit physical access to your system. If you have valuable files on your secure system, someone looking to access them may find that this job is easier if they can walk off with the hard drive and try to get in at their own pace. Things can be made much harder for an attacker if they are kept unaware of the physical aspects of a machine they wish to compromise.

Above all, think beyond the most basic ways to get around your security methods. Consider that you shouldn't protect one possible way to access the system only to leave another avenue far more susceptible. Of course, how you go about doing this is dependent on you or your users' needs. Just be sure not to focus too much on one way in which your system can be attacked.

7.5 The Importance of Secure Passwords

Passwords are the keys to your system. It goes without saying that they should be as secure as possible to prevent an unauthorized login, which is the first step to much bigger security problems. Using passwords that are strong enough to blunt an attack is a crucial yet simple step that can save you a lot of trouble in the future.

Many passwords used by users are quite easy to guess. Red Hat Linux provides a number of different ways to provide authentication to the system, including encrypted passwords using `crypt`, shadow passwords (covered in greater detail in Section 12.1, *Shadow Utilities*), Kerberos 5, and beyond. In every situation where you select a password as part of an authentication scheme, the security of that scheme is at least partially at the mercy of the complexity of the password chosen.

Why should you always try to create secure passwords that are difficult to guess? In short, the price of powerful computer hardware continues to decrease while the number of quality and freely-available tools and methods for cracking passwords continues to increase. Due to the way that passwords are stored in many of the simpler authentication schemes, if an attacker ever gains access to the file containing the passwords of your system's users, they can usually guess one of them in a relatively short amount of time by testing the encrypted passwords against a list of dictionary words. While the authentication schemes are aware of these kinds of attacks and try various methods to help make them less likely, none of these methods is foolproof. Therefore, you should pay great attention to the kind of password you select and how often you change it, especially with the root account.

A good password has the following qualities:

- *Has at least eight characters* — The shorter the password, the generally easier it is to crack.
- *Is made up of characters, numbers, and symbols* — Numbers and symbols hidden within letters (or vice versa) lengthens the possible number of options for a given character, which strengthens the overall password.
- *Is unique* — Select passwords that are different than other passwords you may be using. If all of your passwords are the same or very similar, the magnitude of a security breach can be much greater.

You should avoid using passwords that

- *Are dictionary words* — By using dictionary words as passwords, you are making it exponentially easier for your system to be cracked. Don't do it, and don't override authentication schemes that prevent the use of dictionary words to allow your users to do it.
- *Are tied to your personal information* — If you use passwords that are your birthday, spouse's name, or the make of your car, you are asking for trouble. Think about every password you use and determine whether or not someone who knows you could guess it. If there is even a slight chance they could, don't use that password.
- *Cannot be typed quickly* — If your password is so complicated that you must hunt-and-peck for the characters each time you type it, prying eyes could easily watch your fingers and guess your password. At the very least, practice typing your password while alone to increase the speed in which you can type it.

7.6 Network Security

If you use your Red Hat Linux system on a network (such as a local area network, wide area network, or the Internet), you must be aware that your system is at a greater degree of risk than if you were not connected to that network. Beyond brute attacks on password files and users having inappropriate access, the presence of your system on a larger network widens the opportunity for a security problem and the possible form it may appear.

A number of network security measures have been built into Red Hat Linux, and many open source security tools are also included with the primary distribution. However, despite your preparedness, network security problems may occur, due in part to your network topology or a dozen other factors. To help you determine the source and method of a network security problem, consider the the most likely ways such a problem can occur:

- *Sniffing for authentication data* — Many default authentication methods in Linux and other operating systems depend on sending your authentication information "in the clear," where your username and password is sent over the network in plain text or unencrypted. Tools are widely available for those with access to your network (or the Internet, if you are accessing your system using it) to "sniff" or detect your password by recording all data transferred over the network and sifting through it to find common login statements. This method can be used to find *any* information you send unencrypted, even your root password. It is imperative that you implement and utilize tools like Kerberos 5 and OpenSSH to prevent passwords and other sensitive data form being sent without encryption. If, for whatever reason, these tools cannot be used with your system, then definitely never log in as root unless you are at the console.
- *Frontal attack* — Denial of Service (DoS) attacks and the like can cripple even a secure system by flooding it with improper or malformed requests that overwhelm it or create processes that put your system and its data, as well as other systems that communicate with it, at risk. A number of different protections are available to help stop the attack and minimize the damage, such as packet-filtering firewalls. However, frontal attacks are best handled with a comprehensive look at ways in which untrusted systems communicate with your trusted systems, putting protective barriers between the two, and developing a way to quickly respond to any event so that the disruption and possible damage is limited.
- *Exploiting a security bug or loophole* — Occasionally, bugs are found in software that, if exploited, could do grievous damage to an unprotected system. For that reason, run as few processes as root as possible. Also, use the various tools available to you, such as the Red Hat Network for package updates and security alerts, to fix security problems as soon as they are discovered. Also, make sure that your system has no unnecessary programs starting up at boot time. The fewer programs you have started, the fewer possible security bugs can affect you.

7.7 Additional Resources

Security information is constantly changing, and websites provide a convenient way to get the latest scoop. To stay on top of the recent security notices or to find out more about various security issues involving Red Hat Linux, visit Linux and general security websites regularly. Also, if you need help constructing a solid security policy around the particular needs for your system, use a good security book to give you ideas.

7.7.1 Useful Websites

- http://www.redhat.com/support/errata — Go to the Support section of the Red Hat website for security advisories issued and updates posted for each version of Red Hat Linux by Red Hat.
- http://www.cert.org — The CERT web site offers a very up-to-date list of high-impact security incidents and vulnerabilities, including detailed information on each security notice and how to recover a system after being compromised.
- http://www.sans.org — The System Administration, Networking and Security Institute (SANS) website offers security alerts in a digest form, including convenient links to updated RPMs (when available).
- http://www.linuxsecurity.com — The Linux-Specific Security website has a collection of Linux security-related links, documentation and much more.
- http://www.securityportal.com — The Security Portal website contains a mix of recent security news, Linux-specific fixes, and documents explaining how to construct better security models and policies.

7.7.2 Related Books

- *Securing and Optimizing Linux: Red Hat Edition* by Gerhard Mourani; OpenNA — This book is also available for free download as a PDF file at http://www.openna.com.
- *Secrets & Lies* by Bruce Schneier; John Wiley & Sons, Inc. — A thorough and pragmatic examination of the current computer security issues.

8 Pluggable Authentication Modules (PAM)

Programs that give privileges to users must properly authenticate (verify the identity of) each user. When you log in to a system, you provide your username and password, and the login process uses the username and password to authenticate the login — to verify that you are who you say you are. Forms of authentication other than passwords are possible, and the passwords can be stored in different ways.

Pluggable Authentication Modules (PAM) is a way of allowing the system administrator to set an authentication policy without having to recompile authentication programs. With PAM, you control how particular authentication modules are plugged into a program by editing that program's PAM configuration file in `/etc/pam.d`.

Most Red Hat Linux users will never need to alter PAM configuration files for any of their programs. When you use RPM to install programs that require authentication, they automatically make the changes necessary to do normal password authentication using PAM. However, if you need to customize your configuration, you must understand the structure of a PAM configuration file. More information can be found in Section 8.2.2, *PAM Modules*.

8.1 Advantages of PAM

When used correctly, PAM provides many advantages for a system administrator, such as the following:

- A common authentication scheme that can be used with a wide variety of applications.
- PAM can be implemented with various applications without having to recompile the applications to specifically support PAM.
- Great flexibility and control over authentication for the administrator and application developer.
- Application developers do not need to develop their program to use a particular authentication scheme. Instead, they can focus purely on the details of their program.

8.2 PAM Configuration Files

The directory `/etc/pam.d` contains the PAM configuration files. In earlier versions of PAM, `/etc/pam.conf` was used. The `pam.conf` file is still read if no `/etc/pam.d/` entry is found, but its use is deprecated.

Each application (or *service*, as applications designed to be used by many users are commonly known) has its own file. Each file has five different elements: **service name**, **module type**, **control flag**, **module path**, and **arguments**.

8.2.1 PAM Service Names

The service name of every PAM-enabled application is the name of its configuration file in `/etc/pam.d`. Each program which uses PAM defines its own service name.

For example, the `login` program defines the service name `login`, `ftpd` defines the service name `ftp`, and so on.

In general, the service name is the name of the program used to *access* the service, not the program used to *provide* the service.

8.2.2 PAM Modules

PAM includes four different types of modules for controlling access to a particular service:

- An `auth` module provides the actual authentication (perhaps asking for and checking a password) and sets credentials, such as group membership or Kerberos tickets.
- An `account` module checks to make sure that the authentication is allowed (the account has not expired, the user is allowed to log in at this time of day, and so on).
- A `password` module is used to set passwords.
- A `session` module is used after a user has been authenticated. A `session` module allows someone to use their account (for example, mounting the user's home directory or making their mailbox available).

These modules may be *stacked*, or placed upon one another, so that multiple modules are used. The order of a module stack is very important in the authentication process, because it makes it very easy for an administrator to require that several conditions exist before allowing user authentication to occur.

For example, `rlogin` normally uses at least four stacked authentication methods, as can be seen in its PAM configuration file:

```
auth       required     /lib/security/pam_nologin.so
auth       required     /lib/security/pam_securetty.so
auth       required     /lib/security/pam_env.so
auth       sufficient   /lib/security/pam_rhosts_auth.so
auth       required     /lib/security/pam_stack.so service=system-auth
account    required     /lib/security/pam_stack.so service=system-auth
password   required     /lib/security/pam_stack.so service=system-auth
session    required     /lib/security/pam_stack.so service=system-auth
```

Before someone is allowed to `rlogin`, PAM verifies that the `/etc/nologin` does not exist, that they are not trying to log in remotely as root, and that any environmental variables can be loaded. Then, a successful `rhosts` authentication is performed before the connection is allowed. If `rhosts` authentication fails, then standard password authentication is done.

New PAM modules can be added at any time, and PAM-aware applications can then be made to use them. For example, if you create a one-time-password creation method and write a PAM module to support it, PAM-aware programs can immediately use the new module and password method without being recompiled or otherwise modified in any way. As you can imagine, this is very beneficial, because it lets you mix-and-match, as well as test, authentication methods very quickly with different programs without having to recompile the programs.

Documentation on writing modules is included with the system in `/usr/share/doc/pam-<version-number>`.

8.2.3 PAM Control Flags

All PAM modules generate a success or failure result when checked. Control flags tell PAM what do with the result. Since modules can be stacked in a particular order, control flags give you the ability to set the importance of a module in respect to the modules that follow it.

Again, consider the `rlogin` PAM configuration file:

```
auth       required     /lib/security/pam_nologin.so
auth       required     /lib/security/pam_securetty.so
auth       required     /lib/security/pam_env.so
auth       sufficient   /lib/security/pam_rhosts_auth.so
auth       required     /lib/security/pam_stack.so service=system-auth
account    required     /lib/security/pam_stack.so service=system-auth
password   required     /lib/security/pam_stack.so service=system-auth
session    required     /lib/security/pam_stack.so service=system-auth
```

After the module type is specified, the control flags decide how important that particular module type should be considered to the overall goal of allowing access to the program to that user.

Four types of control flags are defined by the PAM standard:

- `required` flagged modules must be successfully checked in order for the authentication to be allowed. If a `required` module check fails, the user is not notified until any other modules of the same module type have been checked.
- `requisite` flagged modules also must be successfully checked in order for the authentication to be successful. However, if a `requisite` module check fails, the user is notified immediately with a message reflecting the first failed `required` *or* `requisite` module.
- `sufficient` flagged modules checks are ignored if they fail. But, if a `sufficient` flagged module is successfully checked and no `required` flagged modules above it have failed, then

no other modules of this module type are checked and this module type is considered to have successfully been checked as a whole.

- `optional` flagged modules are not crucial for the overall success or failure of that module type's authentication. The only time they play a role is when no other modules of that module type have succeeded or failed. In this case, the success or failure of an `optional` flagged module determines the overall PAM authentication for that module type.

A newer control flag syntax that allows for even more control is now available for PAM. Please see the PAM docs located in `/usr/share/doc/pam–<version-number>` for information on this new syntax.

8.2.4 PAM Module Paths

Module paths tell PAM where to find the pluggable module to be used with the module type specified. Usually, it is provided as the full path to the module, such as `/lib/security/pam_stack.so`. However, if the full path is not given (in other words, the path does not start with a `/`), then the module indicated is assumed to be in `/lib/security`, the default location for PAM modules.

8.2.5 PAM Arguments

PAM uses arguments to pass information to a pluggable module during authentication for a particular module type. These arguments allow the PAM configuration files for particular programs to use a common PAM module but in different ways.

For example, the `pam_userdb.so` module uses secrets stored in a Berkeley DB file to authenticate the user. (Berkeley DB is an open source database system designed to be embedded in many application to track particular types of information.) The module takes a `db` argument, specifying the Berkeley DB filename to use, which can be different for different services.

So, the `pam_userdb.so` line in a PAM configuration file look like this:

```
auth      required  /lib/security/pam_userdb.so db=path/to/file
```

Invalid arguments are ignored and do not otherwise affect the success or failure of the PAM module. When an invalid argument is passed, an error is usually written to `/var/log/messages`. However, as the reporting method is controlled by the PAM module, so it is up to the module to correctly log the error.

8.2.6 PAM Configuration File Samples

A sample PAM application configuration file looks like this:

```
#%PAM-1.0
auth      required  /lib/security/pam_securetty.so
```

```
auth      required  /lib/security/pam_unix.so shadow nullok
auth      required  /lib/security/pam_nologin.so
account   required  /lib/security/pam_unix.so
password  required  /lib/security/pam_cracklib.so
password  required  /lib/security/pam_unix.so shadow nullok use_authtok
session   required  /lib/security/pam_unix.so
```

The first line is a comment (any line starting with a # character is a comment). Lines two through four stack three modules to use for login authentication.

```
auth      required  /lib/security/pam_securetty.so
```

Line two makes sure that *if* the user is trying to log in as root, the tty on which they are logging in is listed in the `/etc/securetty` file, *if* that file exists.

```
auth      required  /lib/security/pam_unix.so shadow nullok
```

Line three causes the user to be asked for a password and the password to be checked.

```
auth      required  /lib/security/pam_nologin.so
```

Line four checks to see if the file `/etc/nologin` exists. If `/etc/nologin` exists and the user is not root, the authentication fails.

Note that all three `auth` modules are checked, *even if the first `auth` module fails*. This strategy prevents the user from knowing why their authentication was not allowed. Knowing why their authentication failed might allow them to break the authentication more easily on their next try. You can change this behavior by changing `required` to `requisite`. If any `requisite` module returns failure, PAM fails immediately without calling any other modules.

```
account   required  /lib/security/pam_unix.so
```

The fifth line causes any necessary account verification to be done. For example, if shadow passwords have been enabled, the `pam_unix.so` module will check to see if the account has expired or if the user has not changed his or her password within the grace period allowed.

```
password  required  /lib/security/pam_cracklib.so
```

The sixth line tests a newly changed password by seeing whether the password can easily be determined by a dictionary-based password cracking program.

```
password  required  /lib/security/pam_unix.so shadow nullok use_authtok
```

The seventh line specifies that if the `login` program changes the user's password, it should use the `pam_unix.so` module to do so. (This will happen only if an `auth` module has determined that the password needs to be changed — for example, if a shadow password has expired.)

```
session   required  /lib/security/pam_unix.so
```

The eighth and final line specifies that the `pam_unix.so` module should be used to manage the session. Currently, that module does not do anything; it could be replaced by any necessary module or supplemented by stacking.

Note that the order of the lines within each file matters. While the order in which `required` modules are called does not matter much, there are other control flags available. While `optional` is rarely used, `sufficient` and `requisite` cause order to become important.

As the next example, we will review the `auth` configuration for `rlogin`:

```
#%PAM-1.0
auth       required     /lib/security/pam_nologin.so
auth       required     /lib/security/pam_securetty.so
auth       required     /lib/security/pam_env.so
auth       sufficient   /lib/security/pam_rhosts_auth.so
auth       required     /lib/security/pam_stack.so service=system-auth
```

First, `pam_nologin.so` checks to see if `/etc/nologin` exists. If is does, no one can log in except for root.

```
auth       required     /lib/security/pam_securetty.so
```

Second, `pam_securetty.so` keeps root logins from occurring on insecure terminals. This effectively disallows all root `rlogin` attempts. If you wish to allow them (in which case you should be behind a good firewall or not be connected to the Internet), see Section 8.4, *Using `rlogin`, `rsh`, and `rexec` with PAM*.

```
auth       required     /lib/security/pam_env.so
```

Third, the `pam_env.so` module loads the environmental variables specified in `/etc/security/pam_env.conf`.

```
auth       sufficient   /lib/security/pam_rhosts_auth.so
```

Fourth, if `pam_rhosts_auth.so` authenticates the user using `.rhosts` in the user's home directory, PAM immediately authenticates the `rlogin` without moving on to do a normal password authentication with `pam_stack.so`. If `pam_rhosts_auth.so` fails to authenticate the user, that failed authentication is ignored.

```
auth       required     /lib/security/pam_stack.so service=system-auth
```

Fifth, if `pam_rhosts_auth.so` has failed to authenticate the user, the `pam_stack.so` module performs normal password authentication, and is passed the `service=system-auth` argument.

Note

If you do not want to prompt for a password when the securetty check fails and determines that the user is trying to login as root remotely, you can change the pam_securetty.so module from required to requisite. Alternatively, if you want to allow root logins remotely (which is not a good idea), you can comment out this line.

8.3 Shadow Passwords

If you are using shadow passwords, pam_unix.so will automatically detect that they are in use and will use them to authenticate users.

Please refer to Section 12.1, *Shadow Utilities* for more information on shadow passwords.

8.4 Using rlogin, rsh, and rexec with PAM

For security reasons, rexec, rsh, and rlogin are not enabled by default in Red Hat Linux 7.1. You should use the OpenSSH suite of tools instead. Information concerning the OpenSSH tools can be found in Chapter 11, *SSH Protocol* and the *Official Red Hat Linux Customization Guide*.

If you must use rexec, rsh, and rlogin, and if you need to use them as root, you will need to make a few modifications to the /etc/securetty file. All three of these tools have PAM configuration files that require the pam_securetty.so PAM module, so you must edit /etc/securetty to allow root access.

Before you can log in as root using these tools, you first have to have them properly set up. First, install the rsh-server RPM, which is included with Red Hat Linux 7.1. See the *Official Red Hat Linux Customization Guide* if you need assistance using RPM.

Next, run ntsysv and enable rexec, rsh, and rlogin. See the ntsysv man page if you need help using this tool.

Finally, restart xinetd with /sbin/service xinetd restart to activate the ntsysv changes. At this point, all users except root will be able to use rexec, rsh, and rlogin.

To allow root to use these tools, add the names of the tools you wish to allow to the /etc/securetty. If you wanted to enable root login using rexec, rsh, and rlogin, add the following lines to /etc/securetty:

```
rexec
rsh
rlogin
```

To allow root to log in using these tools via `telnet` (an even worse idea but necessary in some environments), add a few more lines:

```
pts/0
pts/1
```

8.5 Additional Resources

Much more information about PAM is available than what is covered in this chapter. Various additional sources of information exist and will prove invaluable in helping to configure and use PAM on your system.

8.5.1 Installed Documentation

- `pam` man page — Good introductory information on PAM, including the structure and purpose of the PAM configuration files.
- `/usr/share/doc/pam-<version-number>` — Contains excellent HTML documentation on PAM, including a *System Administrators' Guide*, a *Module Writers' Manual*, and an *Application Developers' Manual*. Also contains a copy of the PAM standard, DCE-RFC 86.0.

8.5.2 Useful Websites

- http://www.kernel.org/pub/linux/libs/pam — The primary distribution website for the Linux-PAM project, containing information on various PAM modules and applications in use or in development, a FAQ, and additional PAM documentation.

In addition to these sources, we suggest that you read as many configuration file examples as possible when beginning to work with PAM. Many websites offer code examples, both for administrators who want to change default configuration files and for application developers who want to use PAM with their programs.

9 Using Kerberos 5 on Red Hat Linux

Kerberos is a secure system for providing network authentication services. Authentication means:

- The identities of entities on the network are verified.
- Traffic on the network is from the source who claims to have sent it.

Kerberos uses passwords to verify the identity of users, and these passwords are always sent over the network in encrypted form.

9.1 Why Use Kerberos?

Most conventional network systems use password-based authentication schemes. When a user needs to authenticate to a service running on a network server, they type in their password for each service that requires authentication. Their password is sent over the network, and the server verifies their identity using the password.

Transmission of passwords in plaintext using this method, while commonly done, is a tremendous security risk. Any system cracker with access to the network and a packet analyzer (also known as a packet sniffer) can intercept any passwords sent this way.

The primary design goal of Kerberos is to ensure that passwords are *never* sent across a network unencrypted and are preferably never sent over the network at all. The proper use of Kerberos will eradicate the threat of packet sniffers intercepting passwords on your network.

9.2 Why Not Use Kerberos?

Kerberos removes a common and severe security threat, so why is it not in use on every network? For several reasons, Kerberos may be difficult to implement:

- No quick solution exists for migrating user passwords from a standard UNIX password database (such as `/etc/passwd` or `/etc/shadow`) to a Kerberos password database. Migration is technically feasible, but this issue is beyond the scope of this chapter. For help deciding whether a password migration makes sense for your Kerberos installation, see the Kerberos FAQ Question 2.23 or the information referenced in Section 9.8, *Additional Resources* for more detailed information concerning this issue.
- Kerberos is only partially-compatible with the Pluggable Authentication Modules (PAM) system used by most servers running Red Hat Linux. For more information on this issue, see Section 9.7, *Kerberos and Pluggable Authentication Modules (PAM)*.
- For an application to use Kerberos, its sources must be modified to make the appropriate calls into the Kerberos libraries. For some applications, this may require too much programming effort.

For other applications, changes must be made to the protocol used between network servers and their clients. Again, this may require extensive programming. Furthermore, it may be impossible to make certain closed-source applications work with Kerberos.

- Kerberos assumes that you are using trusted hosts on an untrusted network. Its primary goal is to prevent plaintext passwords from being sent across that network. However, if anyone other than the proper user has physical access to any of the hosts, especially the one that issues tickets used for authentication, the entire Kerberos authentication system is at risk of being compromised.
- Finally, if you decide to use Kerberos on your network, you must realize that it is an all-or-nothing proposition. If *any* services that transmit plaintext passwords remain in use, passwords can still be compromised, and your network gains no net benefit from the use of Kerberos. To secure your network with Kerberos, you must either **kerberize** (make it work with Kerberos) *all* applications that send plaintext passwords or stop using those insecure applications on your network.

9.3 Kerberos Terminology

Like any other system, Kerberos has its own terminology. Before we talk about how it works, here is a list of terms that you will need to know:

ciphertext

Encrypted data.

client

An entity on the network (a user, a host, or an application) that can get a ticket from Kerberos.

credential cache or ticket file

A file which contains the keys for encrypting communications between a user and various network services. Kerberos 5 provides a framework for using other cache types (such as shared memory), but files are better supported.

key

Data used when encrypting or decrypting other data. Encrypted data cannot be decrypted without the proper key or extremely good guessing.

Key Distribution Center (KDC)

A service that issues Kerberos tickets, usually run on the same host as the Ticket Granting Server.

key table or keytab

A file that includes an unencrypted list of principals and their keys. Servers retrieve the keys they need from keytab files instead of using `kinit`. The default keytab file is

`/etc/krb5.keytab`. The `kadmind` command is the only service that uses any other file (it uses `/var/kerberos/krb5kdc/kadm5.keytab`).

plaintext

Unencrypted data.

principal

A user or service that can authenticate using Kerberos. A principal's name is in the form "*root*[/*instance*]*@REALM*". For a typical user, the *root* is the same as their login ID. The *instance* is optional. If the principal has an instance, it is separated from the root with a forward slash ("/"). An empty string ("") is actually a valid instance (which differs from the default, *NULL* instance), but using it can be confusing. All principals in a realm have their own key, which is derived from their password (for users) or randomly set (for services).

realm

A network that uses Kerberos, composed of one or a few servers (also known as KDCs) and a potentially very large number of clients.

service

A program or computer accessed over the network.

ticket

A temporary set of electronic credentials that verify the identity of a client for a particular service.

Ticket Granting Service (TGS)

Issues tickets for a desired service that are used by the user to actually gain access to the service. The TGS usually runs on the same host as the KDC.

Ticket Granting Ticket (TGT)

A special ticket which allows the client to obtain additional tickets without applying for them from the KDC.

9.4 How Kerberos Works

Now that you have heard a few of the terms that Kerberos uses, here is a simplified explanation of how a Kerberos authentication system works:

On a "normal" network which uses passwords to authenticate users, when a user requests a network service that requires authentication, the user is prompted to type in their password. Their password is transmitted in plaintext over the network, and access to the network service is granted.

As mentioned previously, the central problem solved by Kerberos is how to use passwords for authentication without sending them over the network. On a kerberized network, the Kerberos database contains principals and their keys (for users, their keys are derived from their passwords). The Kerberos database also contains keys for all of the network services.

When a user on a kerberized network logs in to their workstation, their principal is sent to the Key Distribution Center (KDC) as a request for a Ticket Granting Ticket (TGT). This request can be sent by the login program (so that it is transparent to the user) or can be sent by the `kinit` program after the user logs in.

The KDC checks for the principal in its database. If the principal is found, the KDC creates a TGT, encrypts them using the user's key, and sends it back to the user.

The login program or `kinit` decrypts the TGT using the user's key (which it computes from the user's password). The TGT, which is set to expire after a certain period of time, is stored in your credentials cache. An expiration time is set so that a compromised TGT can only be used for a certain period of time, usually eight hours (unlike a compromised password, which could be used until changed). The user won't have to re-enter their password until the TGT expires or they logout and login again.

When the user needs access to a network service, the client uses the TGT to request a ticket for the service from the Ticket Granting Service (TGS), which runs on the KDC. The TGS issues a ticket for the desired service, which is used to authenticate the user.

As you might have guessed, this explanation is a simplified description of Kerberos authentication events. If you need a more in-depth explanation of how Kerberos works, see Section 9.8, *Additional Resources*.

Note

Kerberos depends on certain network services to work correctly. First, Kerberos requires approximate clock synchronization between the machines on your network. If you haven't set up a clock syncing program for your network, you will need to do so. Also, since certain aspects of Kerberos rely on the Domain Name System (DNS), be sure that the DNS entries and hosts on your network are all correctly configured. See the *Kerberos V5 System Administrator's Guide*, provided in PostScript and HTML formats, in `/usr/share/doc/krb5-server-<version-number>`, for more information on these issues.

9.5 Setting Up a Kerberos 5 Server on Red Hat Linux 7.1

When you're setting up Kerberos, install the server(s) first. If you need to set up slave servers, the details of setting up relationships between master and slave servers are covered in the *Kerberos 5 Installation Guide* (in the `/usr/share/doc/krb5-server-<version-number>` directory).

To install a Kerberos server:

1. Be sure that you have clock synchronization and DNS working on your server before installing Kerberos 5. Pay particular attention to time synchronization between the Kerberos server and its various clients. If the server and client clocks are different by more than five minutes (this default amount is configurable in Kerberos 5), Kerberos clients will not be able to authenticate to the server. This clock synchronization is necessary to prevent an attacker from using an old authenticator to masquerade as a valid user.

   You should set up a Network Time Protocol (NTP) compatible client/server network using Red Hat Linux, even if you aren't using Kerberos. Red Hat Linux 7.1 includes the `ntp` package for easy installation. See http://www.eecis.udel.edu/~ntp for additional information on NTP.

2. Install the `krb5-libs`, `krb5-server`, and `krb5-workstation` packages on the dedicated machine which will run your KDC. This machine needs to be secure — if possible, it shouldn't run any services other than the KDC.

   If you'd like to use a Graphical User Interface (GUI) utility to administrate Kerberos, you should also install the `gnome-kerberos` package. It contains `krb5`, a GUI tool for managing tickets, and `gkadmin`, a GUI tool for managing Kerberos realms.

3. Edit the `/etc/krb5.conf` and `/var/kerberos/krb5kdc/kdc.conf` configuration files to reflect your realm name and domain-to-realm mappings. A simple realm can be constructed by replacing instances of *EXAMPLE.COM* and *example.com* with your domain name (be sure to keep uppercase and lowercase names in the correct format) and by changing the KDC from *kerberos.example.com* to the name of your Kerberos server. By convention, all realm names are uppercase and all DNS hostnames and domain names are lowercase. For full details on the formats of these files, see their respective man pages.

4. Create the database using the `kdb5_util` utility from a shell prompt:

   ```
   /usr/kerberos/sbin/kdb5_util create -s
   ```

   The `create` command creates the database that will be used to store keys for your Kerberos realm. The `-s` switch forces creation of a **stash** file in which the master server key is stored. If no stash file is present from which to read the key, the Kerberos server (`krb5kdc`) will prompt the user for the master server password (which can be used to regenerate the key) every time it is started.

5. Edit the `/var/kerberos/krb5kdc/kadm5.acl` file. This file is used by `kadmind` to determine which principals have access to the Kerberos database and their level of access. Most organizations will be able to get by with a single line:

```
*/admin@EXAMPLE.COM  *
```

Most users will be represented in the database by a single principal (with a *NULL*, or empty, instance, such as *joe@EXAMPLE.COM*). With this configuration, users with a second principal with an instance of *admin* (for example, *joe/admin@EXAMPLE.COM*) will be able to wield full power over the realm's Kerberos database.

Once `kadmind` is started on the server, any user will be able to access its services by running `kadmin` or `gkadmin` on any of the clients or servers in the realm. However, only users listed in the `kadm5.acl` file will be able to modify the database in any way, except for changing their own passwords.

Note

The `kadmin` and `gkadmin` utilities communicate with the `kadmind` server over the network, and they use Kerberos to handle authentication. Of course, you need to create the first principal before you can connect to the server over the network to administer it. Create the first principal with the `kadmin.local` command, which is specifically designed to be used on the same host as the KDC and doesn't use Kerberos for authentication.

Type the following `kadmin.local` command at the KDC terminal to create the first principal:

```
/usr/kerberos/sbin/kadmin.local -q "addprinc username/admin"
```

6. Start Kerberos using the following commands:

```
/sbin/service krb5kdc start
/sbin/service kadmin start
/sbin/service krb524 start
```

7. Add principals for your users using the `addprinc` command with `kadmin` or using the **Principal** => **Add** menu option in `gkadmin`. `kadmin` (and `kadmin.local` on the master KDC) is a command line interface to the Kerberos administration system. As such, many commands are available after launching the `kadmin` program. Please see the `kadmin` man page for more information.

8. Verify that your server will issue tickets. First, run `kinit` to obtain a ticket and store it in a credential cache file. Then use `klist` to view the list of credentials in your cache and use `kdestroy` to destroy the cache and the credentials it contains.

Note

By default, `kinit` attempts to authenticate you using the login username of the account you used when you first logged into your system (not the Kerberos server). If that system username does not correspond to a principal in your Kerberos database, you will get an error message. If that happens, just give `kinit` the name of your principal as an argument on the command line (`kinit` *`principal`*).

Once you have completed the steps listed above, your Kerberos server should be up and running. Next, you will need to set up your Kerberos clients.

9.6 Setting Up a Kerberos 5 Client on Red Hat Linux 7.1

Setting up a Kerberos 5 client is less involved than setting up a server. At minimum, you should install the client packages and provide your clients with a valid `krb5.conf` configuration file. Kerberized versions of `rsh` and `rlogin` will also require some configuration changes.

1. Be sure that you have time synchronization in place between the Kerberos client and KDC. See Section 9.5, *Setting Up a Kerberos 5 Server on Red Hat Linux 7.1* for more information. In addition, DNS should be working properly on the Kerberos client before installing the Kerberos client programs.

2. Install the `krb5-libs` and `krb5-workstation` packages on all of the clients in your realm. You must supply your own version of `/etc/krb5.conf` for your client workstations; usually this can be the same `krb5.conf` used by the KDC.

3. Before a particular workstation in your realm can allow users to connect using kerberized `rsh` and `rlogin`, that workstation will need to have the `xinetd` package installed and have its own host principal in the Kerberos database. The `kshd` and `klogind` server programs will also need access to the keys for their service's principal.

   Using `kadmin`, add a host principal for the workstation. The instance in this case will be the hostname of the workstation. Because you'll never need to type the password for this principal again, and you probably don't want to bother with coming up with a good password, you can use the `-randkey` option to kadmin's `addprinc` command to create the principal and assign it a random key:

```
addprinc -randkey host/blah.example.com
```

Now that you have created the principal, you can extract the keys for the workstation by running `kadmin` *on the workstation itself*, and using the `ktadd` command within `kadmin`:

```
ktadd -k /etc/krb5.keytab host/blah.example.com
```

In order to use the kerberized versions of `rsh` and `rlogin`, you must enable `klogin`, `eklogin`, and `kshell`, usually accomplished using `ntsysv` or `chkconfig`.

4. Other kerberized network services will need to be started. To use kerberized `telnet`, you must enable `krb5-telnet`. Use the `ntsysv` or `chkconfig` programs to set the `krb5-telnet` service to start up with your system.

   To provide FTP access, create and extract a key for a principal with a root of ftp, with the instance set to the hostname of the FTP server. Then use `ntsysv` or `chkconfig` to enable `gssftp`.

   The IMAP server included in the `imap` package will use GSS-API authentication using Kerberos 5 if it finds the proper key in `/etc/krb5.keytab`. The root for the principal should be `imap`. The CVS gserver uses a principal with a root of `cvs` and is otherwise identical to a `pserver`.

That should be all you need to do to set up a simple Kerberos realm.

9.7 Kerberos and Pluggable Authentication Modules (PAM)

Currently, kerberized services do not make use of PAM at all — a kerberized server bypasses PAM completely. Applications that use PAM can make use of Kerberos for password checking if the `pam_krb5` module (provided in the `pam_krb5` package) is installed. The `pam_krb5` package contains sample configuration files that will allow services like `login` and `gdm` to authenticate users and obtain initial credentials using their passwords. If access to network servers is always done using kerberized services (or services that use GSS-API, like IMAP), the network can be considered reasonably safe.

Careful system administrators will not add Kerberos password checking to all network services, because most of the protocols used by these services do not encrypt the password before sending it over the network — obviously something to avoid.

9.8 Additional Resources

Kerberos can be a challenge for new users to understand, implement and configure. For more examples and instructions on using Kerberos, refer to the following sources of information:

9.8.1 Installed Documentation

- `/usr/share/doc/krb5-server-<version-number>` — The *Kerberos V5 Installation Guide* and the *Kerberos V5 System Administrator's Guide*, in PostScript and HTML formats, are installed by the `krb5-server` RPM.
- `/usr/share/doc/krb5-workstation-<version-number>` — The *Kerberos V5 UNIX User's Guide*, in PostScript and HTML formats, is installed by the `krb5-workstation` RPM.

9.8.2 Useful Websites

- http://web.mit.edu/kerberos/www — The Kerberos home page on MIT's website.
- http://www.nrl.navy.mil/CCS/people/kenh/kerberos-faq.html — The Kerberos Frequently Asked Questions (FAQ).
- ftp://athena-dist.mit.edu/pub/kerberos/doc/usenix.PS — Link to a PostScript version of *Kerberos: An Authentication Service for Open Network Systems* by Jennifer G. Steiner, Clifford Neuman, and Jeffrey I. Schiller. This document is the original paper describing Kerberos.
- http://web.mit.edu/kerberos/www/dialogue.html — *Designing an Authentication System: a Dialogue in Four Scenes* originally by Bill Bryant in 1988, modified by Theodore Ts'o in 1997. This document is a conversation between two developers who are thinking through the creation of a Kerberos-style authentication system. The conversational style of the discussion make this a good starting place for people who are completely unfamiliar with Kerberos.
- http://www.ornl.gov/~jar/HowToKerb.html — Practical advice on kerberizing your network.

10 Installing and Configuring Tripwire

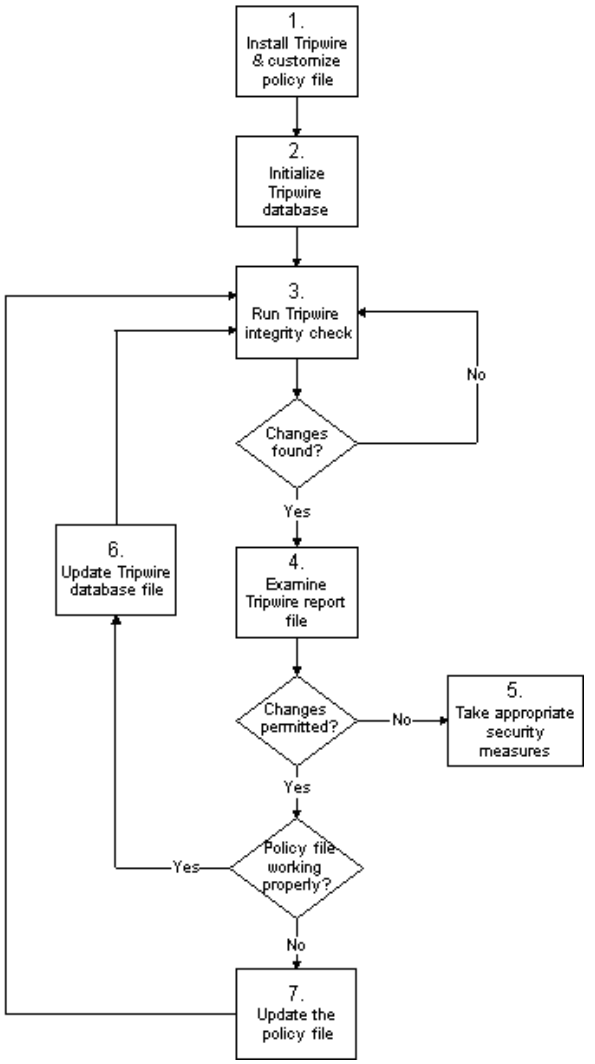
Tripwire software can help to ensure the integrity of critical system files and directories by identifying all changes made to them. Tripwire configuration options include the ability to receive alerts via email if particular files are altered and automated integrity checking via a `cron` job. Using Tripwire for intrusion detection and damage assessment helps you keep track of system changes and can speed the recovery from a break-in by reducing the number of files you must restore to repair the system.

Tripwire compares files and directories against a baseline database of file locations, dates modified, and other data. It generates the baseline by taking a snapshot of specified files and directories in a known secure state. (For maximum security, Tripwire should be installed and the baseline created before the system is at risk from intrusion.) After creating the baseline database, Tripwire compares the current system to the baseline and reports any modifications, additions, or deletions.

10.1 How to Use Tripwire

The following flowchart illustrates how Tripwire should be used:

Figure 10–1 How to Use Tripwire



The following steps should be taken to properly install, use and maintain Tripwire:

1. *Install Tripwire and customize the policy file* — If not already done, install the `tripwire` RPM (see Section 10.2.1, *RPM Installation Instructions*). Then, customize the sample configuration (`/etc/tripwire/twcfg.txt`) and policy (`/etc/tripwire/twpol.txt`) files and run the configuration script (`/etc/tripwire/twinstall.sh`). For more information, see Section 10.2.2, *Post-Installation Instructions*.

2. *Initialize the Tripwire database* — Build a database of critical system files to monitor based on the contents of the new, signed Tripwire policy file (`/etc/tripwire/tw.pol`). For more information, see Section 10.7, *Initializing the Database*.

3. *Run a Tripwire integrity check* — Compare the newly-created Tripwire database with the actual system files, looking for missing or altered files. For more information, see Section 10.8, *Running an Integrity Check*.

4. *Examine the Tripwire report file* — View the Tripwire report file using `twprint` to note integrity violations. For more information, see Section 10.9, *Printing Reports*.

5. *Take appropriate security measures* — If monitored files have been altered inappropriately, you can either replace the originals from backups or reinstall the program.

6. *Update the Tripwire database file* — If the integrity violations are intentional and valid, such as if you intentionally edited a file or replaced a particular program, you should tell Tripwire's database file to not report them as violations in future reports. For more information, see Section 10.10, *Updating the Database after an Integrity Check*.

7. *Update the Tripwire policy file* — If you need to change the list of files Tripwire monitors or how it treats integrity violations, you should update your sample policy file (`/etc/tripwire/twpol.txt`), regenerate a signed copy (`/etc/tripwire/tw.pol`), and update your Tripwire database. For more information, see Section 10.11, *Updating the Policy File*.

Refer to the appropriate sections within this chapter for detailed instructions on these steps.

10.2 Installation Instructions

Once installed, Tripwire must also be correctly initialized to be able to keep a close watch on your files. These sections detail how to install the program, if it is not already present on your system, and then how to initialize the Tripwire database.

10.2.1 RPM Installation Instructions

The easiest way to install Tripwire is to install the `tripwire` RPM during the Red Hat Linux 7.1 installation process. However, if you've already installed Red Hat Linux 7.1, you can use RPM, Gnome-RPM, or Kpackage to install the Tripwire RPM from the Red Hat Linux 7.1 CD-ROMs. The following steps outline this process using RPM:

1. Locate the RedHat/RPMS directory on the Red Hat Linux 7.1 CD-ROM.
2. Locate the tripwire binary RPM by typing ls -l tripwire* in the RedHat/RPMS directory.
3. Type rpm -Uvh *<name>* (where *<name>* is the name of the Tripwire RPM found in step 2)
4. After installing the tripwire RPM, follow the post-installation instructions outlined below.

Note

The release notes and README file are located in /usr/share/doc/tripwire-*<version-number>*. These documents contain important information about the default policy file and other issues.

10.2.2 Post-Installation Instructions

The tripwire RPM installs the program files needed to run the software. After you've installed Tripwire, you must configure it for your system as outlined in the following steps:

1. If you already know of several changes that should be made to the configuration file (/etc/tripwire/twcfg.txt) or the policy file (/etc/tripwire/twpol.txt), edit those files now.

Note

While you should edit your configuration and policy files to customize Tripwire to your particular situation, editing the configuration or policy files is not required to use Tripwire. If you plan to modify the configuration or policy files, you must make these changes before running the configuration script (/etc/tripwire/twinstall.sh). If you modify the configuration or policy files after running the configuration script, you must re-run the configuration script before initializing the database file. Keep in mind that you *can* edit the configuration and policy files *after* initializing the database file and running an integrity check.

---

2. Type /etc/tripwire/twinstall.sh at the command line as root and press [Enter] to run the configuration script. The twinstall.sh script walks you through the processes of setting passphrases, generating the cryptographic keys that protect the Tripwire configuration and policy

files, and signing these files. See Section 10.6, *Selecting Passphrases* for more information on setting passphrases.

Note

Once encoded and signed, the configuration file (`/etc/tripwire/tw.cfg`) and policy file (`/etc/tripwire/tw.pol`) generated by running the `/etc/tripwire/twinstall.sh` script should not be renamed or moved.

---

3. Initialize the Tripwire database file by issuing the `/usr/sbin/tripwire --init` command at the command line.
4. Run the first integrity check comparing your new Tripwire database to your system files by issuing the `/usr/sbin/tripwire --check` command at the command line and looking for errors in the generated report.

Once you finish these steps successfully, Tripwire has the baseline snapshot of your filesystem that it needs to check for changes to critical files. Additionally, the `tripwire` RPM adds a file called `tripwire-check` to the `/etc/cron.daily` directory that will automatically run an integrity check once per day.

10.3 File Locations

Before working with Tripwire, you should know where important files for the application are located. Tripwire stores its files in a variety of places depending on their role:

- The `/usr/sbin` directory stores the `tripwire`, `twadmin`, and `twprint` programs.
- The `/etc/tripwire` directory contains the local and site keys (`*.key` files) and the initialization script (`twinstall.sh`), as well as the sample and actual configuration and policy files.
- The `/var/lib/tripwire` directory contains the Tripwire database of your system's files (`*.twd`) and a `report` directory where Tripwire reports are stored. The Tripwire reports, named *`host_name-date_of_report-time_of_report`*`.twr`, detail the differences between the Tripwire database and your actual system files.

10.4 Tripwire Components

The Tripwire policy file is a text file containing comments, rules, directives, and variables. This file dictates the way Tripwire checks your system. Each rule in the policy file specifies a system object to be monitored. Rules also describe which changes to the object to report and which to ignore.

System objects are the files and directories you wish to monitor. Each object is identified by an object name. A property refers to a single characteristic of an object that Tripwire software can monitor. Directives control conditional processing of sets of rules in a policy file. During installation, the text policy file (`/etc/tripwire/twpol.txt`) is encrypted and renamed, becoming the active policy file (`/etc/tripwire/tw.pol`).

When first initialized, Tripwire uses the signed policy file rules to create the database file (`/var/lib/tripwire/`*`host_name`*`.twd`). The database file is a baseline snapshot of the system in a known secure state. Tripwire compares this baseline against the current system to determine what changes have occurred. This comparison is called an **integrity check**.

When you perform an integrity check, Tripwire produces report files in the `/var/lib/tripwire/report` directory. The report files summarize any file changes that violated the policy file rules during the integrity check.

The Tripwire configuration file (`/etc/tripwire/tw.cfg`) stores system-specific information, such as the location of Tripwire data files. Tripwire generates the necessary configuration file information during installation, but the system administrator can change parameters in the configuration file at any time after that point. Note that the altered configuration file must be signed in the same way as the policy file in order for it to be used by default.

The configuration file variables **POLFILE**, **DBFILE**, **REPORTFILE**, **SITEKEYFILE**, and **LOCALKEYFILE** specify the locations of the policy file, database file, report files, and site and local key files. These variables are defined by default at the time of installation. If you edit the configuration file and leave any of them undefined, the configuration file will be considered invalid by Tripwire. This causes an error on the execution of `tripwire`, making the program exit.

Note that the altered configuration file must be signed in the same way as the policy file in order for it to be used by Tripwire. See Section 10.11.1, *Signing the Configuration File* for instructions on signing the configuration file.

10.5 Modifying the Policy File

You can specify how Tripwire checks your system by modifying the Tripwire policy file (`twpol.txt`). Modifying the policy file to your particular system configuration increases the usefulness of Tripwire reports by minimizing false alerts for files or programs you aren't using but Tripwire is still reporting as altered or missing.

Locate the default policy file at `/etc/tripwire/twpol.txt`. An example policy file (located at `/usr/share/doc/tripwire-<version-number>/policyguide.txt`) is included to help you learn the policy language. Read the example policy file for instructions on how to edit the default policy file.

If you modify the policy file immediately after installing the `tripwire` package, be sure to type `/etc/tripwire/twinstall.sh` to run the configuration script. This script signs the modified policy file and renames it to `tw.pol`. This is the active policy file used by the `tripwire` program when it executes.

If you modify the sample policy file after running the configuration script, see Section 10.11, *Updating the Policy File* for instructions on signing it to make the required `tw.pol` file.

Note

If you modify the sample policy file, it will not be used by Tripwire until it is signed, encrypted and made into the new `/etc/tripwire/tw.pol` file (see Section 10.11, *Updating the Policy File*).

10.6 Selecting Passphrases

Tripwire files are signed or encrypted using site and local keys, which protect the configuration, policy, database, and report files from being viewed or altered except by users who know the site and/or local passphrases. This means that, even if an intruder can obtain root access to your system, they will not be able to alter the Tripwire files to hide their tracks unless they also know the passphrases. When selecting passphrases, you must use at least eight alphanumeric and symbolic characters for each passphrase. The maximum length of a passphrase is 1023 characters. Quotes should not be used as passphrase characters. Also, make sure that your passphrases are completely different from the root password for the system.

You should assign unique passphrases for both the site key and the local key. The site key passphrase protects the site key, which is used to sign Tripwire configuration and policy files. The local key signs Tripwire database and report files.

Store the passphrases in a secure location. *There is no way to decrypt a signed file if you forget your passphrase.* If you forget the passphrases, the files are unusable and you will have to run the configuration script again, which also reinitializes the Tripwire database.

10.7 Initializing the Database

When initializing its database, Tripwire builds a collection of filesystem objects based on the rules in the policy file. This database serves as the baseline for integrity checks.

To initialize the Tripwire database, use the following command:

```
/usr/sbin/tripwire --init
```

This command can take several minutes to run.

10.8 Running an Integrity Check

When running an integrity check, Tripwire compares the current, actual filesystem objects with their properties as recorded in its database. Violations are printed to standard output and saved in a report file that can be accessed later by `twprint`. For more information on viewing Tripwire reports, see Section 10.9, *Printing Reports*.

An email configuration option in the policy file even allows particular email addresses to be sent notices when certain integrity violations occur. See Section 10.12, *Tripwire and Email* for instructions on how to set this up.

To run an integrity check, use the following command:

```
/usr/sbin/tripwire --check
```

This command requires some time to run in most situations, depending upon the number of files to be checked.

10.9 Printing Reports

The `twprint -m r` command will display the contents of a Tripwire report in clear text. You must tell `twprint` which report file to display.

A `twprint` command for printing Tripwire reports looks similar to the following (all on one line):

```
/usr/sbin/twprint -m r --twrfile
          /var/lib/tripwire/report/<name>.twr
```

The `-m r` option in the command tells `twprint` to decode a Tripwire report. The `--twrfile` option tells `twprint` to use a specific Tripwire report file.

The name of the Tripwire report that you want to see includes the name of the host that Tripwire checked to generate the report, plus the creation date and time. You can review previously saved reports at any time. Simply type `ls /var/lib/tripwire/report` to see a list of Tripwire reports.

Tripwire reports can be rather lengthy, depending upon the number of violations found or errors generated. A sample report starts off like this:

```
Tripwire(R) 2.3.0 Integrity Check Report

Report generated by:          root
Report created on:            Fri Jan 12 04:04:42 2001
Database last updated on:     Tue Jan  9 16:19:34 2001

===============================================================================
Report Summary:
===============================================================================
Host name:                    some.host.com
Host IP address:              10.0.0.1
Host ID:                      None
Policy file used:             /etc/tripwire/tw.pol
Configuration file used:      /etc/tripwire/tw.cfg
Database file used:           /var/lib/tripwire/some.host.com.twd
Command line used:            /usr/sbin/tripwire --check

===============================================================================
Rule Summary:
===============================================================================
-------------------------------------------------------------------------------
Section: Unix File System
-------------------------------------------------------------------------------
  Rule Name               Severity Level    Added    Removed  Modified
  ---------               --------------    -----    -------  --------
  Invariant Directories   69                0        0        0
  Temporary directories   33                0        0        0
* Tripwire Data Files     100               1        0        0
  Critical devices        100               0        0        0
  User binaries           69                0        0        0
  Tripwire Binaries       100               0        0        0
```

10.9.1 Using `twprint` to View the Tripwire Database

You can also use `twprint` to view the entire database or information about selected files in the Tripwire database. This is useful for seeing just how much information Tripwire is tracking on your system.

To view the entire Tripwire database, type this command:

```
/usr/sbin/twprint -m d --print-dbfile | less
```

This command will generate a large amount of output, with the first few lines appearing similar to this:

```
Tripwire(R) 2.3.0 Database

Database generated by:        root
Database generated on:        Tue Jan  9 13:56:42 2001
Database last updated on:     Tue Jan  9 16:19:34 2001

===============================================================
Database Summary:
===============================================================
Host name:                    some.host.com
Host IP address:              10.0.0.1
Host ID:                      None
Policy file used:             /etc/tripwire/tw.pol
Configuration file used:      /etc/tripwire/tw.cfg
Database file used:           /var/lib/tripwire/some.host.com.twd
Command line used:            /usr/sbin/tripwire --init

===============================================================
Object Summary:
===============================================================
---------------------------------------------------------------
# Section: Unix File System
---------------------------------------------------------------
     Mode        UID          Size       Modify Time
     ------      ----------   ---------- ----------
 /
     drwxr-xr-x  root (0)     XXX        XXXXXXXXXXXXXXXXX
 /bin
     drwxr-xr-x  root (0)     4096       Mon Jan  8 08:20:45 2001
 /bin/arch
     -rwxr-xr-x  root (0)     2844       Tue Dec 12 05:51:35 2000
 /bin/ash
     -rwxr-xr-x  root (0)     64860      Thu Dec  7 22:35:05 2000
 /bin/ash.static
```

```
-rwxr-xr-x  root (0)     405576     Thu Dec  7 22:35:05 2000
```

To see information about a particular file that Tripwire is tracking, such as `/etc/hosts`, type a different `twprint` command:

```
/usr/sbin/twprint -m d --print-dbfile /etc/hosts
```

The result will look similar to this:

```
Object name:  /etc/hosts

Property:               Value:
-------------           -----------
Object Type             Regular File
Device Number           773
Inode Number            216991
Mode                    -rw-r--r--
Num Links               1
UID                     root (0)
GID                     root (0)
```

See the `twprint` man page for other options.

10.10 Updating the Database after an Integrity Check

If you run an integrity check and Tripwire finds violations, you will first need to determine whether the violations discovered are actual security breaches or the product of authorized modifications. If you recently installed an application or edited critical system files, Tripwire will (correctly) report integrity check violations. In this case, you should update your Tripwire database so those changes are no longer reported as violations. However, if unauthorized changes are made to system files that generate integrity check violations, then you should restore the original file from a backup or reinstall the program.

To update your Tripwire database to accept the violations found in a report, you must specify the report you wish to use to update the database. When issuing the command to integrate those valid violations into your database, be sure to use the most recent report. Type the following command (all on one line), where *name* is the name of the report to be used:

```
/usr/sbin/tripwire --update --twrfile
          /var/lib/tripwire/report/<name>.twr
```

Tripwire will show you the particular report using the default text editor (specified in the Tripwire configuration file on the **EDITOR** line). This is your chance to deselect files that you do not wish to be updated in the Tripwire database. It is important that you only allow authorized integrity violations to be changed in the database.

All proposed updates to the Tripwire database start with a [x] before the file name. If you want to specifically exclude a valid violation from being added to the Tripwire database, remove the x from the box. To accept any files with an x beside them as changes, write the file in the editor and quit the text editor. This signals to Tripwire to alter its database and not report these files as violations.

For example, the default text editor for Tripwire is vi. To write the file with vi and make the changes to the Tripwire database when updating with a specific report, type :wq in vi's command mode and press [Enter]. You will be asked to enter your local passphrase. Then, a new database file will be written to include the valid violations.

After a new Tripwire database is written, the newly authorized integrity violations will no longer show up as warnings when the next integrity check is run.

10.11 Updating the Policy File

If you want to actually change the files Tripwire records in its database or modify the severity in which violations are reported, you need to edit your Tripwire policy file.

First, make whatever changes are necessary to the sample policy file (/etc/tripwire/twpol.txt). A common change to this policy file is to comment out any files that do not exist on your system so that they will not generate a file not found error in your Tripwire reports. For example, if your system does not have a /etc/smb.conf file, you can tell Tripwire not to try to look for it by commenting out its line in twpol.txt:

```
#     /etc/smb.conf                    -> $(SEC_CONFIG) ;
```

Next, you must tell Tripwire to generate a new /etc/tripwire/tw.pol signed file and then generate an updated database file based on this policy information. Assuming /etc/tripwire/twpol.txt is the edited policy file, use this command:

```
/usr/sbin/twadmin --create-polfile -S site.key /etc/tripwire/twpol.txt
```

You will be asked for the site passphrase. Then, the twpol.txt file will be parsed and signed.

It is important that you update the Tripwire database after creating a new /etc/tripwire/tw.pol file. The most reliable way to accomplish this is to delete your current Tripwire database and create a new database using the new policy file.

If your Tripwire database file is named wilbur.domain.com.twd, type this command:

```
rm /var/lib/tripwire/wilbur.domain.com.twd
```

Then type the command to create a new database:

```
/usr/sbin/tripwire --init
```

A new database will be created according to the instructions in the new policy file. To make sure the database was correctly changed, run the first integrity check manually and view the contents of the

resulting report. See Section 10.8, *Running an Integrity Check* and Section 10.9, *Printing Reports* for specific instructions on these points.

10.11.1 Signing the Configuration File

The text file with the configuration file changes (commonly `/etc/tripwire/twcfg.txt`) must be signed to replace the `/etc/tripwire/tw.cfg` and be used by Tripwire when it runs its integrity check. Tripwire will not recognize any configuration changes until the configuration text file is correctly signed and used to replace the `/etc/tripwire/tw.pol` file.

If your altered configuration text file is `/etc/tripwire/twcfg.txt`, type this command to sign it, replacing the current `/etc/tripwire/tw.pol` file:

```
/usr/sbin/twadmin --create-cfgfile -S site.key /etc/tripwire/twcfg.txt
```

Since the configuration file does not not alter any Tripwire policies or files tracked by the application, it is not necessary to regenerate the database of monitored system files.

10.12 Tripwire and Email

Tripwire can email someone if a specific type of rule in the policy file is violated. To configure Tripwire to do this, you first have to know the email address of the person to be contacted if a particular integrity violation occurs, plus the name of the rule you would like to monitor. Note that on large systems with multiple administrators, you can have different sets of people notified for certain violations and no one notified for minor violations.

Once you know who to notify and what to notify them about, add an **emailto=** line to the rule directive section of each rule. Do this by adding a comma after the **severity=** line and putting **emailto=** on the next line, followed by the email addresses to send the violation reports for that rule. Multiple emails will be sent if more than one email address is specified and they are separated by a semi-colon.

For example, if you would like two administrators, Sam and Bob, notified if a networking program is modified, change the Networking Programs rule directive in the policy file to look like this:

```
(
  rulename = "Networking Programs",
  severity = $(SIG_HI),
  emailto = bob@domain.com;sam@domain.com
)
```

Once a new signed policy file is generated from the `/etc/tripwire/twpol.txt` file, the specified email addresses will be notified upon violations of that particular rule. For instructions on signing your policy file, see Section 10.11, *Updating the Policy File*.

10.12.1 Sending Test Email Messages

To make sure that Tripwire's email notification configuration can actually send email correctly, use the following command:

```
/usr/sbin/tripwire --test --email your@email.address
```

A test email will immediately be sent to the email address by the `tripwire` program.

10.13 Additional Resources

Tripwire can do more than what is covered in this chapter. Refer to these additional sources of information to learn more about Tripwire.

10.13.1 Installed Documentation

- `/usr/share/doc/tripwire-<version-number>` — An excellent starting point for learning about how to customize the configuration and policy files in the `/etc/tripwire` directory.
- Also, refer to the man pages for `tripwire`, `twadmin` and `twprint` for help using those utilities.

10.13.2 Useful Websites

- http://www.tripwire.org — The home of the Tripwire Open Source Project, where you can find the latest news on the application, including an FAQ list.

11 SSH Protocol

This chapter covers the benefits of the SSH™ protocol, the sequence of events that occur when a secure connection is made to a remote system, the different layers of SSH, and methods to ensure SSH is used by users connecting to your system.

Common methods for remotely logging into another system through a shell (`telnet`, `rlogin`, or `rsh`) or copying files between hosts (`ftp` or `rcp`) are insecure and should be avoided. Instead, you should only connect to a remote host using a secure shell or an encrypted virtual private network. Using secure methods to remotely log in to other systems will decrease the security risks for both your system and the remote system.

11.1 Introduction

SSH (or *S*ecure *SH*ell) is a protocol for creating a secure connection between two systems. Using SSH, the client machine initiates a connection with a server machine. The following safeguards are provided by SSH:

- After an initial connection, the client can verify that it is connecting to the same server during subsequent sessions.
- The client can transmit its authentication information to the server, such as a username and password, in an encrypted format.
- All data sent and received during the connection is transferred using strong encryption, making it extremely difficult to decrypt and read.
- The client has the ability to use X11[1]applications launched from the shell prompt. This technique provides a secure, graphical interface (called **X11 forwarding**).

The server benefits from SSH, as well, especially if it is running a number of services. If you use **port forwarding**, otherwise insecure protocols (for example, POP) can be encrypted for secure communication with remote machines. SSH makes it relatively simple to encrypt different types of communication normally sent insecurely over public networks.

Red Hat Linux 7.1 includes the OpenSSH server (`openssh-server`) and client (`openssh-clients`) packages, as well as the general OpenSSH package (`openssh`) which must be installed for either of them to work. Please see the *Official Red Hat Linux Customization Guide* for instructions on installing and deploying OpenSSH on your Red Hat Linux system.

[1] X11 refers to the X11R6 windowing display system, traditionally referred to as X. Red Hat Linux includes XFree86, a widely used, open source X Window System, which is based on X11R6.

The OpenSSH packages require the OpenSSL package (`openssl`). OpenSSL installs several important cryptographic libraries that help OpenSSH provide encrypted communications. You must install the `openssl` package before installing any OpenSSH packages.

A large number of client and server programs can use the SSH protocol, including many open source and freely available applications. Several different SSH client versions are available for almost every major operating system in use today. Even if the users connecting to your system are not running Red Hat Linux, they can still find and use an SSH client native for their operating system.

11.1.1 Why Use SSH?

Threats to network traffic include packet sniffing, DNS and IP spoofing[2] and the promulgation of fake routing information. In general terms, these threats can be categorized as follows:

- *Interception of communication between two systems* — In this scenario, a third party exists somewhere on the network between communicating entities and makes a copy of the information being passed between them. The intercepting party may intercept and keep the information, or it may alter the information and send it on to the intended recipient.
- *Impersonation of a particular host* — Using this strategy, an intercepting system pretends to be the intended recipient of a message. If the strategy works, the client remains unaware of the deception and continues to communicate with the interceptor as if its traffic had successfully reached its destination.

Both techniques cause information to be intercepted, possibly for hostile reasons. The results can be disastrous, whether that goal is achieved by listening for all packets on a LAN or a hacked DNS server pointing to a maliciously duplicated host.

If SSH is used for remote shell logins and file copying, these security threats can be greatly diminished. A server's digital signature provides verification for its identity. The entire communication between client and server systems cannot be used if intercepted, because each of the packets is encrypted. Attempts to spoof the identity of either side of a communication will not work, since each packet is encrypted using a key known only by the local and remote systems.

11.2 Event Sequence of an SSH Connection

A certain series of events helps protect the integrity of an SSH communication between two hosts.

First, a secure **transport layer** is created so that the client knows that it is communicating with the correct server. Then, the communication is encrypted between the client and server using a symmetric cipher.

2 Spoofing commonly means appearing to others to be a particular system when you are actually not that system.

Next, with a secure connection to the server in place, the client authenticates itself to the server without worrying that the authentication information may be compromised. OpenSSH on Red Hat Linux uses DSA or RSA keys and version 2.0 of the SSH protocol for authentication by default.

Finally, with the client authenticated to the server, several different services can be safely and securely used through the connection, such as an interactive shell session, X11 applications, and tunneled TCP/IP ports.

The entire connection process occurs with very little extra work required on the local system. In fact, in many respects, SSH works well because it is familiar to users who are accustomed to less secure connection methods.

In the following example, user1 on the client system is initiating an SSH connection to a server. The server's IP address is 10.0.0.2, but its domain name could be used instead. The login name of user1 on the server is user2. The `ssh` command is written as follows:

```
[user1@machine1 user1]$ ssh user2@10.0.0.2
```

The OpenSSH client will request the user's private key passphrase to decrypt the private key, which is used to perform authentication. However, the private key passphrase is not sent across the now secure connection between the client and server. Instead, the passphrase is used to unlock the `id_dsa` file and generate a signature, which it then sends to the server. If the server has a copy of the user's public key which can be used to verify the signature, the user is authenticated.

In this example, the user is using a DSA key (RSA keys, among many others, can also be used) and sees the following prompt:

```
Enter passphrase for DSA key '/home/user1/.ssh/id_dsa':
```

If the public key authentication fails for whatever reason (perhaps the passphrase is entered incorrectly or the authentication information does not already exist on the server), another type of authentication is usually attempted. In our example, the OpenSSH server allows user1 to authenticate herself using user2's password because the signature sent did not match a public key stored by user2:

```
user2@machine2's password:
```

With a correctly entered password, the user is given a shell prompt. Of course, user2 must already have an account on the 10.0.0.2 machine for password authentication to work.

```
Last login: Mon Apr 15 13:27:43 2001 from machine1
[user2@machine2 user2]$
```

At this point, the user can interact with the shell in the same way as they might do with `telnet` or `rsh`, except that the communication is encrypted.

Other SSH tools, `scp` and `sftp`, work in a similar way as the insecure `rcp` and `ftp`, respectively. See the *Official Red Hat Linux Customization Guide* for instructions and examples for using these and other SSH commands.

11.3 Layers of SSH Security

The SSH protocol allows any client and server programs built to the protocol's specifications to communicate securely and be used interchangeably.

Two different varieties of SSH currently exist. SSH version 1 contains several patented encryption algorithms (however, several of these patents have expired) and a security hole that potentially allows for data to be inserted into the data stream. It is recommended that you use SSH version 2-compatible servers and clients, if at all possible.

OpenSSH includes support for version 2 (and freely available DSA encryption keys). Combined with the OpenSSL encryption libraries, OpenSSH provides a full-range of security capabilities.

Both SSH protocol versions (1 and 2) use similar layers of security to strengthen the integrity of the communication from several different angles. Each layer provides its own type of protection, which when used together with the others, strengthens the overall security of the communication and makes it easier to use.

11.3.1 Transport Layer

The primary role of the transport layer is to facilitate safe and secure communication between the two hosts at the time of and after authentication. Usually running over TCP/IP, the transport layer accomplishes this by handling the encryption and decryption of data, verifying that the server is the correct machine for authentication, and providing integrity protection of data packets as they are sent and received. In addition, the transport layer can also provide compression of the data, effectively speeding the transfer of information.

Once a client contacts a server using the SSH protocol, several important points are negotiated so that the two systems can correctly construct the transport layer:

- Key exchange
- The public key algorithm to be used
- The symmetric encryption algorithm to be used
- The message authentication algorithm to be used
- The hash algorithm to be used

During the key exchange, the server identifies itself to the client with a **host key**. Of course, if this client has never communicated with this particular server before, then the server's key will be unknown to the client. OpenSSH gets around this problem by allowing the client to accept the server's host key the first time an SSH connection occurs. Then, in subsequent connections, the server's host key can be checked with a saved version on the client, providing confidence that the client is indeed communicating with the intended server.

The host key verification method used by OpenSSH is not perfect. An attacker could masquerade as the server during the initial contact, as the local system would not necessarily know the difference between the intended server and the attacker at that point. But, until a better host key distribution method becomes widely available, this initially insecure method is better than nothing.

SSH is designed to work with almost any kind of public key algorithm or encoding format. After an initial key exchange creates two values (a hash value used for exchanges and a shared secret value), the two systems immediately begin calculating new keys and algorithms to protect authentication and future data sent over the connection.

11.3.2 Authentication

Once the transport layer has constructed a secure tunnel to pass information between the two systems, the server tells the client the different authentication methods supported, such as using a private key-encoded signature or typing a password. The client will then try to authenticate itself to the server using any of the supported methods.

Since servers can be configured to allow different types of authentication, this method gives each side the optimal amount of control. The server can decide which encryption methods it will support based on its security model, and the client can choose the order of authentication methods to attempt from among the available options. Thanks to the secure nature of the SSH transport layer, even seemingly insecure authentication methods, such as a host-based authentication, are safe to use.

Most users requiring a secure shell will authenticate using a password. Unlike other security authentication schemes, the password is transmitted to the server in cleartext. However, since the entire password is encrypted when moving over the the transport layer, it can be safely sent across any network.

11.3.3 Connection

After a successful authentication over the SSH transport layer, multiple **channels** are opened by multiplexing[3]the single connection between the two systems. Each of these channels handles communication for a different terminal session, forwarded X11 information, or any other separate service seeking to use the SSH connection.

[3] A multiplexed connection consists of several signals being sent over a shared, common medium. With SSH, different channels are sent over a common secure connection.

Both clients and servers can create a new channel, with each channel being assigned a different number at each end. When one side attempts to open a new channel, that side's number for the channel is sent along with the request. This information is stored by the other side and used to direct a particular type of service's communication to that channel. This is done so that different types of sessions will not affect one another and channels can be closed without disrupting the primary SSH connection between the two systems.

Channels also support flow-control, which allows them to send and receive data in an orderly fashion. In this way, data is not sent over the channel until the host receives a message that the channel is able to receive it.

Channels are particularly useful with X11 forwarding and TCP/IP port forwarding with SSH. Separate channels can be configured differently, perhaps to use a different maximum packet size or to transfer a particular type of data. This allows SSH to be flexible in handling different types of remote connections, such as dial-up over public networks or high speed LAN links, without having to change the basic infrastructure of the protocol. The client and server negotiate the configuration of each channel within the SSH connection for the user automatically.

11.4 OpenSSH Configuration Files

OpenSSH has two different sets of configuration files, one for the client programs (`ssh`, `scp`, and `sftp`) and the other for the server service (`sshd`), located in two different areas.

System-wide SSH configuration information is stored in the `/etc/ssh` directory:

- `primes` — Contains Diffie-Hellman groups used for the Diffie-Hellman key exchange. Basically, this key exchange creates a shared secret value that cannot be determined by either party alone and is used to provide host authentication. This file is critical for constructing a secure transport layer.
- `ssh_config` — The system-wide SSH client configuration file used to direct the SSH client. If a user has her own configuration file available in her home directory (`~/.ssh/config`), then its values will override the values stored in `/etc/ssh/ssh_config`.
- `sshd_config` — The configuration file for `sshd`.
- `ssh_host_dsa_key` — The DSA private key used by `sshd`.
- `ssh_host_dsa_key.pub` — The DSA public key used by `sshd`.
- `ssh_host_key` — The RSA private key used by `sshd` for version 1 of the SSH protocol.
- `ssh_host_key.pub` — The RSA public key used by `sshd` for version 1 of the SSH protocol.
- `ssh_host_rsa_key` — The RSA private key used by `sshd` for version 2 of the SSH protocol.

- `ssh_host_rsa_key.pub` — The RSA public key used by `sshd` for version 2 of the SSH protocol.

User-specific SSH configuration information is stored in the user's home directory within the `.ssh` subdirectory:

- `authorized_keys2` — The file that holds a list of "authorized" public keys. If a connecting user can prove that she knows the private key which corresponds to any of these, then she is authenticated. Note that this is only an optional authentication method.
- `id_dsa` — Contains the DSA authentication identity of the user.
- `id_dsa.pub` — The DSA public key of the user.
- `known_hosts2` — Stores the DSA host keys of the servers a user logs into via SSH when the user elects to record them. If a server has its host keys legitimately altered, perhaps on a re-installation of Red Hat Linux, the user will be notified that the host key stored in the `known_hosts2` file that corresponds with this host does not match. Then, the user must delete that host's key in `known_hosts` in order to store the new host key for that system. The `known_hosts2` file is very important for ensuring that the client is connecting the correct server. If a host's key has changed, and you are not absolutely certain why it has changed, then you should contact the host's system administrator to make sure that the host has not be compromised.

See the man pages for `ssh` and `sshd` for information concerning the various directives available in the SSH configuration files.

11.5 More Than a Secure Shell

A secure command line interface is just the beginning of the many ways SSH can be used. Given the proper amount of bandwidth, X11 sessions can be directed over an SSH channel. Or, by using TCP/IP forwarding, previously insecure port connections between systems can be mapped to specific SSH channels.

11.5.1 X11 Forwarding

Opening an X11 session over an established SSH connection is as easy as running an X program while already running an X client on your host. When an X program is run from the secure shell prompt, the SSH client and server create a new secure channel within the current SSH connection, and the X program data is sent over that channel to your client machine as if you were connected to the X server via a local terminal.

As you might imagine, X11 forwarding can be very useful. For example, you can use X11 forwarding to create a secure, interactive session with the `up2date` GUI on the server to selectively update packages (if you have the necessary Red Hat Network packages installed on the server). To do this, simply connect to the server using `ssh` and type:

```
up2date
```

You will be asked to supply the root password for the server. Then, the Red Hat Update Agent will appear and you can update your packages on the server as though you were sitting in front of the machine.

The processing overhead required to encrypt and decrypt the secure information being sent over the channel, plus the extra bandwidth necessary to send encrypted X application data, may be significant, however. Adequate testing is required to make sure that the X program is still usable, given your particular hardware and bandwidth conditions.

11.5.2 TCP/IP Forwarding

TCP/IP forwarding works with the SSH client requesting that a particular port on the client or server side be mapped over the existing SSH connection.

To map a local port on the client to a remote port on the server, you first have to know the port numbers on both machines. It is even possible to map two non-standard, different ports to each other.

To create a TCP/IP forwarding channel which listens for connections on the local host, use the following command (all on one line):

```
ssh -L <local-port>:<remote-hostname>:<remote-port>
        <username>@<hostname>
```

Note

Setting up TCP/IP forwarding to listen on ports below 1024 requires root access, just as starting services that listen on ports below 1024.

For example, if you want to check your email on a server called mail.domain.com using POP and SSH is available on that server, you can use this command to set up TCP/IP forwarding:

```
ssh -L 1100:mail.domain.com:110 mail.domain.com
```

After the TCP/IP forwarding is in place between the two machines, you can direct your POP mail client to use localhost as the POP server and 1100 as the port to check for new mail. Any requests sent to port 1100 on your system will be directed securely to the mail.domain.com server.

If mail.domain.com is not running an SSH server daemon but you can log in via SSH to a machine near it, perhaps through a firewall, you can still use SSH to secure the part of the POP connection that occurs over public networks. A slightly different command is needed:

```
ssh -L 1100:mail.domain.com:110 other.domain.com
```

In this example, you are forwarding your POP request from port 1100 on your machine through the SSH connection on port 22 to other.domain.com. Then, other.domain.com connects to port 110 on mail.domain.com to allow you to check for new mail. Only the connection between your system and other.domain.com is secure, but in many situations, this is enough to get your information safely through public networks by providing more security than you had before.

Of course, in this example and the one above it, you must be able authenticate to the SSH server to perform the TCP/IP forwarding. Be sure that you can execute normal SSH commands before attempting to set up TCP/IP forwarding.

TCP/IP forwarding can be particularly useful for getting information securely through network firewalls. If the firewall is configured to allow SSH traffic via its standard port (22) but block access through other ports, a connection between two hosts using the blocked ports is still possible by redirecting their communication over an established SSH connection between them.

Note

This can be very dangerous, however. Using TCP/IP forwarding to forward connections in this manner allows any user on the client system to connect to the service you are forwarding connections to, which can be hazardous if your client system becomes compromised.

Check with the system administrator who administers your firewall before using TCP/IP forwarding to bypass it. System administrators concerned about TCP/IP forwarding can disable this functionality on the server by specifying a `No` parameter for the **AllowTcpForwarding** line in `/etc/ssh/sshd_config` and restarting the `sshd` service.

11.6 Requiring SSH for Remote Connections

For SSH to be truly effective in protecting your network connections, you must stop using all insecure connection protocols, such as `telnet` and `rsh`. Otherwise, a user's password may be protected using `ssh` on one day only to be captured when they log in the next day using `telnet`.

To disable insecure connection methods to your system, use `ntsysv` or `chkconfig` to make sure that these services do not start up with the system. To use `ntsysv` to configure services that start at runlevels 2, 3, and 5, type the command:

```
/usr/sbin/ntsysv 235
```

Within `ntsysv`, you can disable services from starting up by deselecting them. The [Spacebar] toggles a service between being active or inactive. At a minimum, you should deselect `telnet`, `rsh`,

`ftp`, and `rlogin`. When finished, select the **OK** button to save your `ntsysv` changes. See the `ntsysv` man page for additional assistance using this utility.

Changes made to with `ntsysv` will not take affect until either the system is restarted or changes run-levels. If you disabled services used with `xinetd`, you must restart `xinetd`. By default, `rlogin`, `rsh`, and `telnet` are controlled by `xinetd`. To restart `xinetd`, type:

```
/sbin/service xinetd restart
```

For services not used with `xinetd`, you must stop them manually unless you are restart your system after using `ntsysv`. To stop a service, you will probably use a command such as:

```
/sbin/service <service-name> stop
```

After restarting `xinetd` and stopping any other services you have configured not to start up automatically, disabled connection methods will no longer be accepted by your system. If you disable all remote connection methods other than the `sshd` service daemon, users will have to use an SSH client application to connect to the server.

12 Controlling Access and Privileges

System security relies heavily on users or groups not being able to do more than they should, according to a common security policy. Most of the day-to-day changes concerned with controlling access and privileges revolves around properly using users and groups. (See Chapter 2, *Users and Groups* for more information on properly creating and configuring users and groups.)

However, many organizations using Red Hat Linux have particular guidelines or work environments that require tighter security or special configurations for enhanced or restricted access to applications or system devices. This section discusses a few ways you can tweak your system to provide an appropriate level of access and privileges for your users based on your situation.

12.1 Shadow Utilities

If you are in a multiuser environment and not using PAM or Kerberos, you should consider using Shadow Utilities (also known as **shadow passwords**) for the enhanced protection offered for your system's authentication files. During the installation of Red Hat Linux, shadow password protection for your system is enabled by default, as are **MD5 passwords** (an alternative and arguably more secure method of encrypting passwords for storage on your system).

Shadow passwords offer a few distinct advantages over the previous standard of storing passwords on UNIX and Linux systems, including:

- Improved system security by moving the encrypted passwords (normally found in `/etc/passwd`) to `/etc/shadow` which is readable only by root
- Information concerning password aging (how long it has been since a password was last changed)
- Control over how long a password can remain unchanged before the user is required to change it
- The ability to use the `/etc/login.defs` file to enforce a security policy, especially concerning password aging

The `shadow-utils` package contains a number of utilities that support:

- Conversion from normal to shadow passwords and back (`pwconv`, `pwunconv`)
- Verification of the password, group, and associated shadow files (`pwck`, `grpck`)
- Industry-standard methods of adding, deleting and modifying user accounts (`useradd`, `usermod`, and `userdel`)
- Industry-standard methods of adding, deleting, and modifying user groups (`groupadd`, `groupmod`, and `groupdel`)
- Industry-standard method of administering the `/etc/group` file using `gpasswd`

Note

There are some additional points of interest concerning these utilities:

- The utilities will work properly whether shadowing is enabled or not.
- The utilities have been slightly modified to support Red Hat's user private group scheme. For a description of the modifications, see the `useradd` man page. For more information on user private groups, turn to Section 2.4, *User Private Groups*.
- The `adduser` script has been replaced with a symbolic link to `/usr/sbin/useradd`.
- The tools in the `shadow-utils` package are not Kerberos or LDAP enabled. New users will be local only. For more information on Kerberos and LDAP, see Chapter 9, *Using Kerberos 5 on Red Hat Linux* and Chapter 4, *Lightweight Directory Access Protocol (LDAP)*.

12.2 Configuring Console Access

When normal (non-root) users log in to a computer locally, they are given two types of special permissions:

1. They can run certain programs that they would not otherwise be able to run
2. They can access certain files (normally special device files used to access diskettes, CD-ROMs, and so on) that they would not otherwise be able to access

Since there are multiple consoles on a single computer and multiple users can be logged into the computer locally at the same time, one of the users has to "win" the race to access the files. The first user to log in at the console owns those files. Once the first user logs out, the next user who logs in will own the files.

In contrast, *every* user who logs in at the console will be allowed to run programs that accomplish tasks normally restricted to the root user. If X is running, these actions can be included as menu items in a graphical user interface. As shipped, the console-accessible programs include `halt`, `poweroff` and `reboot`.

12.2.1 Disabling Shutdown Via Ctrl-Alt-Del

By default, `/etc/inittab` specifies that your system is set to shutdown and reboot the system in response to a [Ctrl]-[Alt]-[Del] key combination used at the console. If you'd like to completely disable this ability, you will need to comment out the following line in `/etc/inittab`:

```
ca::ctrlaltdel:/sbin/shutdown -t3 -r now
```

Alternatively, you may just want to allow certain non-root users the right to shutdown the system from the console using [Ctrl]-[Alt]-[Del]. You can restrict this privilege to certain users, by taking the following steps:

1. Add a `-a` option to the `/etc/inittab` line shown above, so that it reads:

   ```
   ca::ctrlaltdel:/sbin/shutdown -a -t3 -r now
   ```

   The `-a` flag tells `shutdown` to look for the `/etc/shutdown.allow` file, which you'll create in the next step.

2. Create a file named `shutdown.allow` in `/etc`. The `shutdown.allow` file should list the usernames of any users who are allowed to shutdown the system using [Ctrl]-[Alt]-[Del]. The format of the `/etc/shutdown.allow` file is a list of usernames, one per line, like the following:

   ```
   stephen
   jack
   sophie
   ```

According to this example `shutdown.allow` file, stephen, jack, and sophie are allowed to shutdown the system from the console using [Ctrl]-[Alt]-[Del]. When that key combination is used, the `shutdown -a` in `/etc/inittab` checks to see if any of the users in `/etc/shutdown.allow` (or root) are logged in on a virtual console. If one of them is, the shutdown of the system will continue; if not, an error message will be written to the system console instead.

For more information on `shutdown.allow` see the `shutdown` man page.

12.2.2 Disabling Console Program Access

In order to disable access by users to console programs, you should run this command as root:

```
rm -f /etc/security/console.apps/*
```

In environments where the console is otherwise secured (BIOS and LILO passwords are set, [Ctrl]-[Alt]-[Delete] is disabled, the power and reset switches are disabled, and so forth), you may not want to allow any user at the console to run `poweroff`, `halt`, and `reboot`, which are accessible from the console by default.

To remove these abilities, run the following commands as root:

```
rm -f /etc/security/console.apps/poweroff
rm -f /etc/security/console.apps/halt
rm -f /etc/security/console.apps/reboot
```

12.2.3 Disabling All Console Access

The PAM `pam_console.so` module manages console file permissions and authentication. (See Chapter 8, *Pluggable Authentication Modules (PAM)* for more information on configuring PAM.) If you want to disable all console access, including program and file access, comment out all lines that refer to `pam_console.so` in the `/etc/pam.d` directory. The following script will do the trick:

```
cd /etc/pam.d
for i in * ; do
sed '/[^#].*pam_console.so/s/^/#/' < $i > foo && mv foo $i
done
```

12.2.4 Defining the Console

The `pam_console.so` module uses the `/etc/security/console.perms` file to determine the permissions for users at the system console. The syntax of the file is very flexible; you can edit the file so that these instructions no longer apply. However, the default file has a line that looks like this:

```
 <console>=tty[0-9][0-9]* :[0-9]\.[0-9] :[0-9]
```

When users log in, they are attached to some sort of named terminal, either an X server with a name like `:0` or `mymachine.example.com:1.0` or a device like `/dev/ttyS0` or `/dev/pts/2`. The default is to define that local virtual consoles and local X servers are considered local, but if you want to consider the serial terminal next to you on port `/dev/ttyS1` to also be local, you can change that line to read:

```
<console>=tty[0-9][0-9]* :[0-9]\.[0-9] :[0-9] /dev/ttyS1
```

12.2.5 Making Files Accessible From the Console

In `/etc/security/console.perms`, there is a section with lines like:

```
<floppy>=/dev/fd[0-1]* \
    /dev/floppy/*
<sound>=/dev/dsp* /dev/audio* /dev/midi* \
    /dev/mixer* /dev/sequencer \
    /dev/sound/*
<cdrom>=/dev/cdrom* /dev/cdwriter*
```

You can add your own lines to this section, if necessary. Make sure that any lines you add refer to the appropriate device. For example, you could add the following line:

```
<scanner>=/dev/sga
```

(Of course, make sure that `/dev/sga` is really your scanner and not, say, your hard drive.)

That's the first step. The second step is to define what is done with those files. Look in the last section of `/etc/security/console.perms` for lines similar to:

```
<console> 0660 <floppy> 0660 root.floppy
<console> 0600 <sound>  0640 root
<console> 0600 <cdrom>  0600 root.disk
```

and add a line like:

```
<console> 0600 <scanner> 0600 root
```

Then, when you log in at the console, you will be given ownership of the `/dev/sga` device and the permissions will be 0600 (readable and writable by you only). When you log out, the device will be owned by root and still have 0600 (now: readable and writable by root only) permissions.

12.2.6 Enabling Console Access for Other Applications

If you wish to make other applications accessible to console users, you will have to do just a little bit more work.

First of all, console access *only* works for applications which reside in `/sbin` or `/usr/sbin`, so the application that you wish to run must be there. After verifying that, do the following steps:

1. Create a link from the name of your application, such as our sample *`foo`* program, to the `/usr/bin/consolehelper` application:

   ```
   cd /usr/bin
   ln -s consolehelper foo
   ```

2. Create the file `/etc/security/console.apps/`*`foo`*:

   ```
   touch /etc/security/console.apps/foo
   ```

3. Create a PAM configuration file for the *`foo`* service in `/etc/pam.d/`. An easy way to do this is to start with a copy of the halt service's PAM configuration file, and then modify the file if you want to change the behavior:

   ```
   cp /etc/pam.d/halt /etc/pam.d/foo
   ```

Now, when you run `/usr/bin/`*`foo`*, it will call `consolehelper`, which will authenticate the user with the help of `/usr/sbin/userhelper`. To authenticate the user, `consolehelper` will ask for the user's password if `/etc/pam.d/`*`foo`* is a copy of `/etc/pam.d/halt` (otherwise, it

will do precisely what is specified in `/etc/pam.d/`*`foo`*) and then run `/usr/sbin/`*`foo`* with root permissions.

12.3 The `floppy` Group

If, for whatever reason, console access is not appropriate for you and you need to give non-root users access to your system's diskette drive, this can be done using the `floppy` group. Simply add the user(s) to the `floppy` group using the tool of your choice. Here's an example showing how `gpasswd` can be used to add user fred to the `floppy` group:

```
[root@bigdog root]# gpasswd -a fred floppy
Adding user fred to group floppy
[root@bigdog root]#
```

Now, user fred will now be able to access the system's diskette drive.

Part III Apache-Related Reference

13 Using Apache as a Secure Web Server

13.1 Introduction

This chapter provides basic information on how to install the Apache World Wide Web (WWW or Web) server with the mod_ssl security module and the OpenSSL library and toolkit. The combination of these three components, provided with Red Hat Linux, will be referred to in this manual as the secure Web server or just as the secure server.

Simply stated, Web servers provide Web pages in response to requests from browsers. Well-known browsers include Netscape Navigator and Microsoft Internet Explorer. In more technical terms, Web servers and browsers communicate using the HyperText Transfer Protocol (HTTP), the Internet standard for Web communications. When users click on a link on a Web page, an HTTP request is sent to a Web server for the content named by the link. The Web server receives the request and provides the content that was asked for, such as a HyperText Markup Language (HTML) page, a CGI script, or a Web page dynamically generated from a database. If a Web server cannot fulfill the request, it sends back an error message. Apache, the Web server provided in Red Hat Linux, is the most widely used Web server on the Internet today (see http://www.netcraft.net/survey).

The Apache Web server is modular in design; it consists of many separate pieces of code which apply to different aspects or functionalities of the Web server. This modularity was intentional, so that any developer can write their own small piece of code to address a particular need. Their code, called a module, can then be integrated into the Apache Web server with relative ease.

The mod_ssl module is a security module for the Apache Web server. The mod_ssl module uses the tools provided by the OpenSSL Project to add a very important feature to Apache — the ability to encrypt communications. In contrast, using regular HTTP, communications between a browser and a Web server are sent in plaintext, which could be intercepted and read by someone along the route between the browser and the server.

The OpenSSL Project includes a toolkit which implements the Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols and a general purpose cryptography library. The SSL protocol is used for secure data transmission over the Internet today. The TLS protocol is a proposed Internet standard for private and reliable communications over the Internet. OpenSSL tools are used by the mod_ssl module to provide security for Web communications.

This chapter is not meant to be complete and exclusive documentation for any of these programs. When possible, this guide will point you to appropriate places where you can find more in-depth documentation on particular subjects.

This chapter will show you how to install these programs. You will also learn the steps necessary to generate a private key and a certificate request, how to generate your own self-signed certificate, and how to install a certificate to use with your secure Web server.

13.2 Acknowledgments

The secure Web server includes the following:

- Software developed by the Apache Group for use in the Apache HTTP server project (http://httpd.apache.org)
- The mod_ssl security module, developed by Ralf S. Engelschall (http://www.modssl.org)
- The OpenSSL toolkit, developed by Mark J. Cox, Ralf S. Engelschall, Dr. Stephen Henson, and Ben Laurie (http://www.openssl.org)
- Software based on the Apache-SSL HTTP server project developed by Ben Laurie (http://www.apache-ssl.org)
- Software based on SSLeay cryptographic software written by Eric Young and Tim Hudson

Red Hat gratefully acknowledges these contributions.

13.3 An Overview of Security-Related Packages

To install the secure server, you will need to install three packages at minimum:

`apache`

The `apache` package contains the `httpd` daemon and related utilities, configuration files, icons, Apache modules, man pages and other files used by the Apache Web server.

`mod_ssl`

The `mod_ssl` package includes the mod_ssl module, which provides strong cryptography for the Apache Web server via the Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols.

`openssl`

The `openssl` package contains the OpenSSL toolkit. The OpenSSL toolkit implements the SSL and TLS protocols and also includes a general purpose cryptography library.

Additionally, other software packages included with Red Hat Linux can provide certain security functionalities (but are not required by the secure server to function):

`apache-devel`

The `apache-devel` package contains the Apache include files, header files and the APXS utility. You will need all of these if you intend to load any extra modules, other than the modules provided with this product. Please see Section 14.3, *Adding Modules to Your Server* for more information on loading modules into your secure Web server using Apache's DSO functionality.

If you do not intend to load other modules into your secure Web server, you do not need to install this package.

apache-manual

The apache-manual package contains the Apache Project's *Apache 1.3 User's Guide* in HTML format. This manual is also available on the Web at http://httpd.apache.org/docs/.

OpenSSH packages

The OpenSSH packages provide the OpenSSH set of network connectivity tools for logging in to and executing commands on a remote machine. OpenSSH tools encrypt all traffic (including passwords), so you can avoid eavesdropping, connection hijacking, and other attacks on the communications between your machine and the remote machine.

The openssh package includes core files needed by both the OpenSSH client programs and the OpenSSH server. The openssh package also contains scp, a secure replacement for rcp (for copying files between machines) and ftp (for transferring files between machines).

The openssh-askpass package supports the display of a dialog window which prompts for a password during use of the OpenSSH agent with RSA authentication.

The openssh-askpass-gnome package contains a GNOME GUI desktop environment dialog window which is displayed when OpenSSH programs prompt for a password. If you are running GNOME and using OpenSSH utilities, you should install this package.

The openssh-server package contains the sshd secure shell daemon and related files. The secure shell daemon is the server side of the OpenSSH suite, and must be installed on your host if you want to allow SSH clients to connect to your host.

The openssh-clients package contains the client programs needed to make encrypted connections to SSH servers, including the following: ssh, a secure replacement for rsh; and slogin, a secure replacement for rlogin (for remote login) and telnet (for communicating with another host via the TELNET protocol).

For more information about OpenSSH, see Chapter 11, *SSH Protocol* and the OpenSSH website at http://www.openssh.com.

openssl-devel

The openssl-devel package contains the static libraries and the include file needed to compile applications with support for various cryptographic algorithms and protocols. You need to install this package only if you are developing applications which include SSL support — you do not need this package to use SSL.

stunnel

The `stunnel` package provides the Stunnel SSL wrapper. Stunnel supports the SSL encryption of TCP connections, so it can provide encryption for non-SSL aware daemons and protocols (such as POP, IMAP and LDAP) without requiring any changes to the daemon's code.

Table 13–1, *Security Packages* displays the location of the secure server packages and additional security-related packages within the package groups provided by Red Hat Linux. This table also tells you whether each package is optional or not for the installation of a secure Web server.

Table 13–1 Security Packages

Package Name	Located in Group	Optional?
`apache`	System Environment/Daemons	no
`mod_ssl`	System Environment/Daemons	no
`openssl`	System Environment/Libraries	no
`apache-devel`	Development/Libraries	yes
`apache-manual`	Documentation	yes
`openssh`	Applications/Internet	yes
`openssh-askpass`	Applications/Internet	yes
`openssh-askpass-gnome`	Applications/Internet	yes
`openssh-clients`	Applications/Internet	yes
`openssh-server`	System Environment/Daemons	yes
`openssl-devel`	Development/Libraries	yes
`stunnel`	Applications/Internet	yes

13.4 How to Install the Secure Server

You can install the secure Web server in the following ways:

- *During a fresh installation of Red Hat Linux* — Since the secure Web server is included with the Red Hat Linux operating system, the easiest method is during the installation of Red Hat Linux. If you are about to do a complete installation of Red Hat Linux, this is how you should install your secure server. See Section 13.5, *Installing the Secure Server with Red Hat Linux* for more information on this method.

- *When you are upgrading Red Hat Linux using the installation program* — if you already have a previous version of Red Hat Linux running on your system and you are upgrading to Red Hat Linux 7.1, you can install the secure server packages during the upgrade process. See Section 13.6, *Upgrading from a Previous Version of Red Hat Linux* for more information on this method.
- *Installing the secure server after installing Red Hat Linux 7.1* — if you previously installed Red Hat Linux 7.1 but did not install the secure server packages, and at a later date decide that you want to install the secure server, you can use RPM, Gnome-RPM or Kpackage to install the secure server packages from a Red Hat Linux CD. See Section 13.7, *Installing the Secure Server After Red Hat Linux* for instructions on how to install the secure server after you have already installed Red Hat Linux.

Upgrading Apache

When you install the secure Web server, if you are upgrading from any previous version of Apache (including any previous secure server product from Red Hat), you will need to understand certain issues concerning the Apache upgrade process. If you are upgrading Apache, see Section 13.8, *Upgrading from a Previous Version of Apache* before you begin the installation process.

13.5 Installing the Secure Server with Red Hat Linux

If you are installing Red Hat Linux and the secure Web server at the same time, follow the instructions provided in the installation manual appropriate for your architecture. If you are planning to use your Red Hat Linux system as a secure server, you will probably choose to perform a server- or custom-class installation. The various installation classes to choose from are as follows:

- If you choose a server-class installation, the `apache`, `mod_ssl` and `openssl` packages will be selected automatically. The `stunnel` and `openssh` packages, which provide security-related functionalities, will also be selected.
- If you choose a workstation-class installation (or a laptop-class installation, if available for your system), the secure server packages and the security-related packages will not be automatically selected for installation, but you can choose to install them during the package selection customization process.
- If you choose a custom-class installation, since you have complete control over what packages are installed, you will need to select the secure server packages and any security-related packages you want.

Once you have chosen an installation class, continue following the installation instructions to partition and configure your system. When you reach the section on selecting package groups, or components,

select the **Web Server** package group. **Web Server** includes the `apache` and `mod_ssl` packages that you must install to run the secure server. Since `openssl` is a dependency for the `mod_ssl` package, `openssl` will also be chosen for installation.

If you would like to install any of the additional security-related packages described in Section 13.3, *An Overview of Security-Related Packages*, you will need to identify their packages to the installation program. To do this, choose **Select individual packages** on the same **Package Group Selection** screen.

Select the security-related packages that you want to install according to the instructions provided in your installation manual. To help you find them, a table of their locations is provided as Table 13–1, *Security Packages*.

After making sure that the packages you need are selected, continue with the installation process. When you have finished installing Red Hat Linux and the secure server, see Section 13.9, *An Overview of Certificates and Security*.

13.6 Upgrading from a Previous Version of Red Hat Linux

If you are already running a previous version of Red Hat Linux on your system, you may choose to upgrade to Red Hat Linux 7.1 instead of performing a full installation. If you decide to upgrade, you must choose **Upgrade** instead of choosing an installation class. Follow the instructions on how to upgrade your system contained in the installation manual appropriate for your architecture. During the upgrade, you will need to make sure that the secure server packages are selected by the installation program.

When you perform an upgrade to your Red Hat Linux system, the installation program checks to see what packages are already installed. Those packages will automatically be updated to the versions included in Red Hat Linux 7.1 during the upgrade process. However, if you do not have a particular package installed, the installation program will not install the new version of that package — unless you customize your upgrade.

If you are upgrading from Red Hat Linux 7.0 or later and had the secure Web server packages installed, the upgrade process will upgrade the secure server packages. If you are upgrading from Red Hat Linux 7.0 or later, but you did not have the secure Web server packages installed, you will need to select the `apache`, `mod_ssl` and `openssl` packages during the package customization process. See Section 13.6.1, *Customizing Your Upgrade to Install the Secure Server* for instructions on finding the packages you will need to choose.

If you are upgrading from the US/Canada version of Red Hat Linux Professional, you will need to customize your upgrade and choose the secure server packages for installation. You may already have `apache` installed, but `mod_ssl` and `openssl` will not be installed, as they were not included in

Red Hat Linux before Red Hat Linux 7.0. You will need to customize the upgrade to choose at least `mod_ssl` and `openssl`. See Section 13.6.1, *Customizing Your Upgrade to Install the Secure Server* for instructions on finding the packages you will need to choose.

If you are upgrading from the International version of Red Hat Linux Professional and had the `apache`, `mod_ssl` and `openssl` packages installed, then the installation program will select and upgrade these programs automatically.

If you are upgrading from the International version of Red Hat Linux Professional but did not have the `apache`, `mod_ssl` or `openssl` packages installed, then you will need to customize your upgrade and choose these packages for installation. See Section 13.6.1, *Customizing Your Upgrade to Install the Secure Server* for instructions on finding the packages you will need to choose.

13.6.1 Customizing Your Upgrade to Install the Secure Server

If you need to customize the upgrade process, follow the upgrading instructions contained in your installation manual; basically, choose **Upgrade** as your **Install Type** and then select **Customize packages to be upgraded**. Then you will need to select the packages to upgrade, as described in your installation manual. To help you select packages, Table 13–1, *Security Packages* provides the location of each secure server-related package and whether it is optional.

After you have finished, if you are also upgrading any version of Apache, see Section 13.8, *Upgrading from a Previous Version of Apache*. If you are not upgrading Apache, continue on to Section 13.9, *An Overview of Certificates and Security*.

13.7 Installing the Secure Server After Red Hat Linux

If you installed Red Hat Linux 7.1 without installing the secure server packages and then, at a later date, decide that you want to install the secure server, you can. The easiest way to do this is to use RPM, Gnome-RPM, or Kpackage to install the RPM packages included on the Red Hat Linux CD.

13.7.1 Stop Any Running Web Server Processes

Before you begin this process, if you are running any Web server on your system, you must stop the server process before installing the secure Web server. If you are running an Apache Web server, stop the server process by issuing one or both of the following commands:

```
/etc/rc.d/init.d/httpsd stop
/etc/rc.d/init.d/httpd stop
```

13.7.2 Using Gnome-RPM or Kpackage

If you are running GNOME or KDE, you can use a GUI program like Gnome-RPM or Kpackage to install the secure server packages.

More information on how to use Gnome-RPM is included in the *Official Red Hat Linux Getting Started Guide*. Instructions on how to use Kpackage are included on the *Kpackage Handbook* Web page at http://www.general.uwa.edu.au/u/toivo/kpackage.

After you have installed the necessary packages, the next step is to create your key and obtain a certificate. Please continue to Section 13.9, *An Overview of Certificates and Security*.

13.7.3 Using RPM

The secure Web server packages are provided in RPM format, so you can install the packages using RPM. See the *Official Red Hat Linux Customization Guide* for more information about RPM. See Table 13–1, *Security Packages* if you are not sure which packages to install.

After you have installed the secure server packages, if you are upgrading any version of Apache, please see Section 13.8, *Upgrading from a Previous Version of Apache*. If you are not upgrading Apache, continue to Section 13.9, *An Overview of Certificates and Security*.

13.8 Upgrading from a Previous Version of Apache

During the installation of the secure server packages, if you are upgrading Apache, you will need to be aware of two issues:

- In the version of Apache included in Red Hat Linux 7.1, the `DocumentRoot` is `/var/www/html`.
- If you customized your Apache configuration file (`httpd.conf`), you will want to know what will happen to your customizations during the upgrade process.

13.8.1 Where is the DocumentRoot?

Basically, the `DocumentRoot` is the directory on your system which holds most of the Web pages served by your Apache Web server. The `DocumentRoot` is set by a configuration directive in Apache's configuration file, `httpd.conf`. If you are unfamiliar with the `DocumentRoot` configuration directive, see Section 14.2.28, *`DocumentRoot`* for a more detailed explanation.

Before Red Hat Linux 7.0, the Apache provided with Red Hat Linux used `/home/httpd/html` as the `DocumentRoot`. In the default (non-secure) version of Apache's configuration file, the `DocumentRoot` is `/usr/local/apache/htdocs`. It is also possible that you (or a predecessor) used an entirely different `DocumentRoot`. In Red Hat Linux 7.1, however, the default `DocumentRoot` is `/var/www/html`.

Does this matter to you? It does, if you used Apache with a different `DocumentRoot`, and you want to serve those same Web pages with your new configuration of Apache. Any Web pages that were previously served from a different `DocumentRoot` will not be found (or served) by the Apache shipped with Red Hat Linux 7.1 in its default configuration. You will need to take one of the following steps:

Move all of the files in your old `DocumentRoot` (`/home/httpd/html`, `/usr/local/apache/htdocs`, or wherever) to the new `DocumentRoot` (`/var/www/html`).

or

Edit the Apache configuration file and change all references to the `DocumentRoot` back to the old directory path.

The solution you choose depends upon your system's configuration. Generally, if you automount `/home` on your system, you will not want to have your `DocumentRoot` in `/home`. On the other hand, if you do not have much space in `/var`, then you probably will not want your `DocumentRoot` in `/var`. You, or your system administrator, will have to decide the best solution based on your system's configuration and your Web server's needs. The secure Web server's default configuration is intended to address the needs of most Webmasters. Unfortunately, we cannot configure it for every individual situation.

13.8.2 What Happens to My Old Configuration File?

If you had another version of Apache installed and you customized its configuration file or files, the configuration files will be saved in their directory with an extension of `.rpmsave` during the installation of Apache. If you had another version of Apache installed but you never altered its configuration file(s), they will be written over during the installation of this product.

After installing Apache, you can cut and paste the customizations from your old Apache configuration file(s) into the newly installed `httpd.conf` configuration file for your secure server. Please note that if you are going to use the Apache Configuration Tool, you must not edit `httpd.conf` by hand. Please see the *Official Red Hat Linux Customization Guide* for more information about the Apache Configuration Tool.

13.9 An Overview of Certificates and Security

Your secure Web server provides security using a combination of the Secure Sockets Layer (SSL) protocol and (in most cases) a digital certificate from a Certificate Authority (CA). SSL handles the encrypted communications and the mutual authentication between browsers and your secure Web server. The CA-approved digital certificate provides authentication for your secure Web server (the CA puts its reputation behind its certification of your organization's identity). When your browser is communicating using SSL encryption, you will see the https:// prefix at the beginning of the Uniform Resource Locator (URL) in the navigation bar.

Encryption depends upon the use of keys (think of them as secret encoder/decoder rings in data format). In conventional or symmetric cryptography, both ends of the transaction have the same key, which they use to decode each other's transmissions. In public or asymmetric cryptography, two keys co-exist: a public key and a private key. A person or an organization keeps their private key a secret, and publishes their public key. Data encoded with the public key can only be decoded with the private key; data encoded with the private key can only be decoded with the public key.

To set up your secure server, you will use public cryptography to create a public and private key pair. In most cases, you will send your certificate request (including your public key), proof of your company's identity, and payment to a CA. The CA will verify the certificate request and your identity, and then send back a certificate for your secure Web server.

A secure server uses a certificate to identify itself to Web browsers. You can generate your own certificate (called a "self-signed" certificate) or you can get a certificate from a Certificate Authority or CA. A certificate from a reputable CA guarantees that a website is associated with a particular company or organization.

Alternatively, you can create your own self-signed certificate. Note, however, that self-signed certificates should not be used in most production environments. Self-signed certificates will not be automatically accepted by a user's browser — the user will be asked by the browser if they want to accept the certificate and create the secure connection. See Section 13.11, *Types of Certificates* for more information on the differences between self-signed and CA-signed certificates.

Once you have a self-signed certificate or a signed certificate from the CA of your choice, you will need to install it on your secure Web server.

13.10 Using Pre-Existing Keys and Certificates

If you already have an existing key and certificate (for example, if you are installing the secure Web server to replace another company's secure Web server product), you will probably be able to use your existing key and certificate with the secure Web server. In the following two situations, you will not be able to use your existing key and certificate:

- *If you are changing your IP address or domain name* — You can not use your old key and certificate if you are changing your IP address or domain name. Certificates are issued for a particular IP address and domain name pair. You will need to get a new certificate if you are changing your IP address or domain name.
- *If you have a certificate from VeriSign and you are changing your server software* — VeriSign is a widely used CA. If you already have a VeriSign certificate for another purpose, you may have been considering using your existing VeriSign certificate with your new secure Web server. However, you will not be allowed to, because VeriSign issues certificates for one particular server software and IP address/domain name combination.

If you change either of those parameters (for example, if you previously used another secure Web server product and now you want to use the secure Web server), the VeriSign certificate you obtained to use with the previous configuration will not work with the new configuration. You will need to obtain a new certificate.

If you have an existing key and certificate that you can use, you will not have to generate a new key and obtain a new certificate. However, you may need to move and rename the files which contain your key and certificate.

Move your existing key file to:

```
/etc/httpd/conf/ssl.key/server.key
```

Move your existing certificate file to:

```
/etc/httpd/conf/ssl.crt/server.crt
```

After you have moved your key and certificate, skip to Section 13.15, *Testing Your Certificate*.

If you are upgrading from the Red Hat Secure Web Server versions 1.0 and 2.0, your old key (`httpsd.key`) and certificate (`httpsd.crt`) will be located in `/etc/httpd/conf/`. You will need to move and rename your key and certificate, so that the secure Web server can use them. Use the following two commands to move and rename your key and certificate files:

```
mv /etc/httpd/conf/httpsd.key /etc/httpd/conf/ssl.key/server.key
mv /etc/httpd/conf/httpsd.crt /etc/httpd/conf/ssl.crt/server.crt
```

Then start your secure Web server as described in Section 14.1, *Starting and Stopping `httpd`*. You should not need to get a new certificate, if you are upgrading from a previous version of the secure Web server.

13.11 Types of Certificates

If you installed your secure Web server using the Red Hat Linux installation program, a random key and a test certificate are generated and put into the appropriate directories. Before you begin using your secure server, however, you will need to generate your own key and obtain a certificate which correctly identifies your server.

You need a key and a certificate to operate your secure Web server — which means that you can either generate a self-signed certificate or purchase a CA-signed certificate from a CA. What are the differences between the two?

A CA-signed certificate provides two important capabilities for your server:

- Browsers will (usually) automatically recognize the certificate and allow a secure connection to be made, without prompting the user.

- When a CA issues a signed certificate, they are guaranteeing the identity of the organization that is providing the Web pages to the browser.

If your secure server is being accessed by the public at large, your secure Web server needs a certificate signed by a CA, so that people who visit your website can rely that the website is owned by the organization who claims to own it. Before signing a certificate, a CA verifies that the organization requesting the certificate was actually who they claimed to be.

Most Web browsers that support SSL have a list of CAs whose certificates they will automatically accept. If a browser encounters a certificate whose authorizing CA is not in the list, the browser will ask the user to choose whether to accept or decline the connection.

You can generate a self-signed certificate for your secure Web server, but be aware that a self-signed certificate will not provide the same functionality as a CA-signed certificate. A self-signed certificate will not be automatically recognized by users' browsers, and a self-signed certificate does not provide any guarantee concerning the identity of the organization that is providing the website. A CA-signed certificate provides both of these important capabilities for a secure server. If your secure server will be used in a production environment, you will probably need a CA-signed certificate.

The process of getting a certificate from a CA is fairly easy. A quick overview is as follows:

1. Create an encryption private and public key pair.
2. Create a certificate request based on the public key. The certificate request contains information about your server and the company hosting it.
3. Send the certificate request, along with documents proving your identity, to a CA. We cannot tell you which certificate authority to choose. Your decision may be based on your past experiences, or on the experiences of your friends or colleagues, or purely on monetary factors.

   To see a list of CAs, click on the **Security** button on your Navigator toolbar or on the padlock icon at the bottom left of the screen, then click on **Signers** to see a list of certificate signers from whom your browser will accept certificates. You can also search the Web for CAs. Once you have decided upon a CA, you will need to follow the instructions they provide on how to obtain a certificate from them.
4. When the CA is satisfied that you are indeed who you claim to be, they will send you a digital certificate.
5. Install this certificate on your Web server, and begin handling secure transactions.

Whether you are getting a certificate from a CA or generating your own self-signed certificate, the first step is to generate a key. See Section 13.12, *Generating a Key* for instructions on how to generate a key.

13.12 Generating a Key

First, `cd` to the `/etc/httpd/conf` directory. Remove the fake key and certificate that were generated during the installation with the following commands:

```
rm ssl.key/server.key
rm ssl.crt/server.crt
```

Next, you need to create your own random key. Type in the following command:

```
make genkey
```

Your system will display a message similar to the following:

```
umask 77 ; \
/usr/bin/openssl genrsa -des3 1024 > /etc/httpd/conf/ssl.key/server.key
Generating RSA private key, 1024 bit long modulus
.......++++++
.....................................................................++++++
e is 65537 (0x10001)
Enter PEM pass phrase:
```

You now need to type in a password. For best security, your password should contain at least eight characters, include numbers and/or punctuation, and not be a word in a dictionary. Also, remember that your password is case sensitive.

Note

You will need to remember and enter this password every time you start your secure Web server, so do not forget it.

You will be asked to re-type the password, to verify that it is correct. Once you have typed it in correctly, a file called `server.key`, containing your key, will be created.

Note that if you do not want to type in a password every time you start your secure Web server, you will need to use the following two commands instead of `make genkey` to create the key. Both of these commands should be typed in entirely on one line.

Use the following command:

```
/usr/bin/openssl genrsa 1024 > /etc/httpd/conf/ssl.key/server.key
```

to create your key. Then use this command:

```
chmod go-rwx /etc/httpd/conf/ssl.key/server.key
```

to make sure that the permissions are set correctly on your key.

After you use the above commands to create your key, you will not need to use a password to start your secure Web server.

Disabling the password feature for your secure Web server is a security risk. We DO NOT recommend that you disable the password feature for your secure Web server.

The problems associated with not using a password are directly related to the security maintained on the host machine. For example, if an unscrupulous individual compromises the regular UNIX security on the host machine, that person could obtain your private key (the contents of your `server.key` file). The key could be used to serve Web pages that will appear to be from your Web server.

If UNIX security practices are rigorously maintained on the host computer (all operating system patches and updates are installed as soon as they are available, no unnecessary or risky services are operating, and so on), the secure Web server's password may seem unnecessary. However, since your secure Web server should not need to be re-booted very often, the extra security provided by entering a password is a worthwhile effort in most cases.

The `server.key` file should be owned by the root user on your system and should not be accessible to any other user. Make a backup copy of this file and keep the backup copy in a safe, secure place. You need the backup copy because if you ever lose the `server.key` file after using it to create your certificate request, your certificate will no longer work and the CA will not be able to help you. Your only option would be to request (and pay for) a new certificate.

If you are going to purchase a certificate from a CA, continue to Section 13.13, *Generating a Certificate Request to Send to a CA*. If you are generating your own self-signed certificate, continue to Section 13.14, *Creating a Self-Signed Certificate*.

13.13 Generating a Certificate Request to Send to a CA

Once you have created a key, the next step is to generate a certificate request which you will need to send to the CA of your choice. Type in the following command:

```
make certreq
```

Your system will display the following output and will ask you for your password (unless you disabled the password option):

```
umask 77 ; \
```

```
/usr/bin/openssl req -new -key /etc/httpd/conf/ssl.key/server.key
-out /etc/httpd/conf/ssl.csr/server.csr
Using configuration from /usr/share/ssl/openssl.cnf
Enter PEM pass phrase:
```

Type in the password that you chose when you were generating your key. Your system will display some instructions and then ask for a series of responses from you. Your inputs will be incorporated into the certificate request. The display, with example responses, will look like this:

```
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) [AU]:US
State or Province Name (full name) [Some-State]:North Carolina
Locality Name (eg, city) []:Durham
Organization Name (eg, company) [Internet Widgits]:Test Company
Organizational Unit Name (eg, section) []:Testing
Common Name (your name or server's hostname) []:test.mydomain.com
Email Address []:admin@mydomain.com
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

The default answers appear in brackets `[]` immediately after each request for input. For example, the first information required is the name of the country where the certificate will be used, shown like the following:

```
Country Name (2 letter code) [AU]:
```

The default input, in brackets, is **AU**. To accept the default, just press [Enter], or fill in your country's two letter code.

You will have to type in the rest of the inputs (`State or Province Name`, `Locality Name`, `Organization Name`, `Organizational Unit Name`, `Common Name`, and `Email address`). All of these should be self-explanatory, but you need to follow these guidelines:

- Do not abbreviate the locality or state. Write them out (for example, St. Louis should be written out as Saint Louis).

- If you are sending this CSR to a CA, be very careful to provide correct information for all of the fields, but especially for the `Organization Name` and the `Common Name`. CAs check the information provided in the CSR to determine whether your organization is responsible for what you provided as the `Common Name`. CAs will reject CSRs which include information they perceive as invalid.
- For `Common Name`, make sure you type in the *real* name of your secure Web server (a valid DNS name) and not any aliases which the server may have.
- The `Email Address` should be the email address for the webmaster or system administrator.
- Avoid any special characters like @, #, &, !, etc. Some CAs will reject a certificate request which contains a special character. So, if your company name includes an ampersand (&), spell it out as "and" instead of "&."
- Do not use either of the extra attributes (`A challenge password` and `An optional company name`). To continue without entering these fields, just press [Enter] to accept the blank default for both inputs.

When you have finished entering your information, a file named `server.csr` will be created. This file is your certificate request, ready to send to your CA.

After you have decided on a CA, follow the instructions they provide on their website. Their instructions will tell you how to send your certificate request, any other documentation that they require, and your payment to them.

After you have fulfilled the CA's requirements, they will send a certificate to you (usually by email). Save (or cut and paste) the certificate that they send you as `/etc/httpd/conf/ssl.crt/server.crt`.

13.14 Creating a Self-Signed Certificate

You can create your own self-signed certificate. Please note that a self-signed certificate will not provide the security guarantees provided by a CA-signed certificate. See Section 13.11, *Types of Certificates* for more details about certificates.

If you would like to make your own self-signed certificate, you will first need to create a random key using the instructions provided in Section 13.12, *Generating a Key*. Once you have a key, use the following command:

```
make testcert
```

You will see the following output and you will be prompted for your password (unless you generated a key without a password):

```
umask 77 ; \
/usr/bin/openssl req -new -key /etc/httpd/conf/ssl.key/server.key
```

```
-x509 -days 365 -out /etc/httpd/conf/ssl.crt/server.crt
Using configuration from /usr/share/ssl/openssl.cnf
Enter PEM pass phrase:
```

After you enter your password (or without a prompt if you created a key without a password), you will be asked for more information. The computer's output and a set of inputs looks like the following (you will need to provide the correct information for your organization and host):

```
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) [AU]:US
State or Province Name (full name) [Some-State]:North Carolina
Locality Name (eg, city) []:Durham
Organization Name (eg, company) [Internet Widgits]:My Company, Inc.
Organizational Unit Name (eg, section) []:Documentation
Common Name (your name or server's hostname) []:myhost.mydomain.com
Email Address []:myemail@mydomain.com
```

After you provide the correct information, a self-signed certificate will be created and placed in `/etc/httpd/conf/ssl.crt/server.crt`. You will need to restart your secure server after generating the certificate. See Section 14.1, *Starting and Stopping `httpd`* for instructions on restarting your secure Web server.

13.15 Testing Your Certificate

When the secure server is installed by the Red Hat Linux installation program, a random key and a generic certificate are installed, for testing purposes. You can connect to your secure server using this certificate. For any purposes other than testing, however, you need to get a certificate from a CA or generate a self-signed certificate. See Section 13.11, *Types of Certificates* if you need more information on the different types of certificates available.

If you have purchased a certificate from a CA or generated a self-signed certificate, you should have a file named `/etc/httpd/conf/ssl.key/server.key`, containing your key, and a file named `/etc/httpd/conf/ssl.crt/server.crt`, containing your certificate. If your key and certificate are somewhere else, move them to these directories. If you changed any of the default locations or filenames for the secure Web server in your Apache configuration files, you should put these two files in the appropriate directory, based on your modifications.

Now, stop and start your server as described in Section 14.1, *Starting and Stopping* `httpd`. If your key file is encrypted, you will be asked for the password. Type in your password and your server should start.

Point your Web browser to your server's home page. The URL to access your secure Web server will look like this:

```
https://your_domain
```

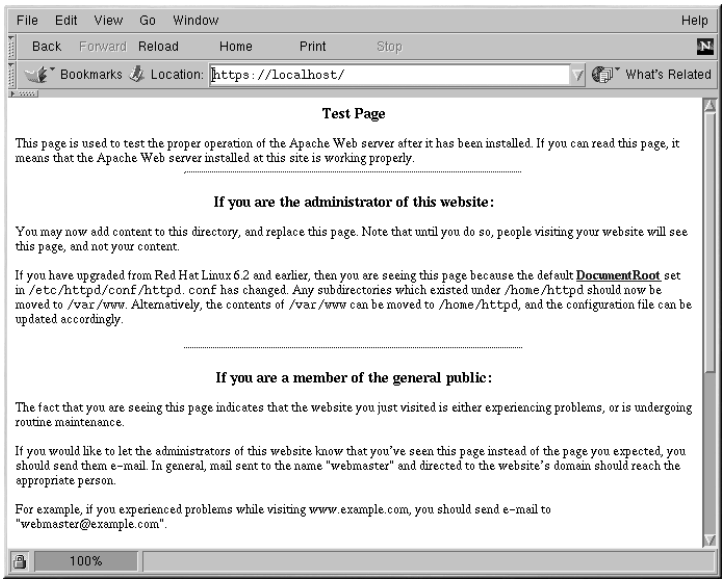
Note

Note the "s" after "http." The https: prefix is used for secure HTTP transactions.

If you are using a CA-signed certificate from a well-known CA, your browser will probably automatically accept the certificate (without prompting you for input) and create the secure connection. Your browser will not automatically recognize a test or a self-signed certificate, because the certificate is not signed by a CA. If you are not using a certificate from a CA, follow the instructions provided by your browser to accept the certificate. You can just accept the defaults by clicking **Next** until the dialogs are finished.

Once your browser accepts the certificate, your secure Web server will show you a default home page as shown in Figure 13–1, *The Default Home Page*.

Figure 13–1 The Default Home Page



13.16 Accessing Your Secure Server

To access your secure server, use a URL like this:

```
https://your_domain
```

Note that URLs which are intended to connect to your secure Web server should begin with the https: protocol designator instead of the more common http: protocol designator.

Your non-secure server can be accessed using an URL like this:

```
http://your_domain
```

The standard port for secure Web communications is port 443. The standard port for non-secure Web communications is port 80. The secure Web server default configuration listens on both of the two standard ports. Therefore, you will not need to specify the port number in a URL (the port number is assumed).

However, if you configure your server to listen on a non-standard port (i.e., anything besides 80 or 443), you will need to specify the port number in every URL which is intended to connect to the server on the non-standard port.

For example, you may have configured your server so that you have a virtual host running non-secured on port 12331. Any URLs intended to connect to that virtual host must specify the port number in the URL. The following URL example will attempt to connect to a non-secure Web server listening on port 12331:

```
http://your_domain:12331
```

Some of the example URLs used in this manual may need to be changed, depending upon whether you are accessing your secure Web server or your non-secure Web server. Please view all URLs in this manual as general examples and not as explicit instructions that will work under all circumstances.

13.17 Additional Resources

If you followed the steps outlined in Chapter 13, *Using Apache as a Secure Web Server* but you experienced a problem, the first thing you should do is check the Red Hat Errata section of the Red Hat website at http://www.redhat.com/support/errata.

If you purchased an Official Red Hat product which included support, you are entitled to technical support. Be sure to visit the Red Hat Support website at http://www.redhat.com/support to register for support.

You may want to subscribe to the redhat-secure-server mailing list. You can subscribe to this mailing list at http://www.redhat.com/mailing-lists.

You can also subscribe to the redhat-secure-server mailing list by emailing `redhat-secure-server-request@redhat.com` and include the word "subscribe" (without the quotation marks) in the Subject line.

13.17.1 Installed Documentation

If you installed the `apache-manual` package, you can access Apache documentation in HTML format on your machine from the following URL: http://localhost/manual/.

The mod_ssl documentation is provided at the following URL: http://localhost/manual/mod/mod_ssl/.

13.17.2 Useful Websites

Tips, FAQs and HOWTO documents are provided on the Red Hat website at http://www.redhat.com/support/docs/howto.

The Red Hat Linux Apache Centralized Knowledgebase is available at http://www.redhat.com/support/docs/faqs/RH-apache-FAQ/book1.html.

The Apache website provides complete documentation for the Apache Web server at http://httpd.apache.org/docs.

The mod_ssl website (http://www.modssl.org) is the definitive source for information about mod_ssl. The website includes a wealth of documentation, including a *User Manual* at http://www.modssl.org/docs.

13.17.3 Related Books

Apache: The Definitive Guide, 2nd edition, by Ben Laurie and Peter Laurie, O'Reilly & Associates, Inc.

14 Apache Directives and Modules

Apache's default configuration should work for most users. You may never need to change any of Apache's configuration directives. If you do want to change any of the default configuration options, you will need to know what some of the options are and where to find them. This chapter covers the configuration options available to you.

WARNING

If you plan to use the Apache Configuration Tool, a GUI utility provided with Red Hat Linux, you must not edit your Apache Web server's `httpd.conf` configuration file. Conversely, if you want to edit `httpd.conf` by hand, do not use the Apache Configuration Tool.

If you need more information on the Apache Configuration Tool, please see the *Official Red Hat Linux Customization Guide*.

After you have installed the `apache` package, the Apache Web server's documentation is available at http://your_domain/manual/ or you can use the Apache documentation available on the Web at http://httpd.apache.org/docs/. The Apache Web server's documentation contains a full list and complete descriptions of all of Apache's configuration options. For your convenience, this chapter provides short descriptions of the configuration directives used in the version of Apache provided with Red Hat Linux.

When you are reading your Web server's configuration file, be aware that it includes both a non-secure and a secure Web server. The secure Web server runs as a virtual host, which is also configured in the `httpd.conf` file. For more information about virtual hosts, see Section 14.4, *Using Virtual Hosts*.

Note

We do not include FrontPage extensions, because the Microsoft™ license prohibits the inclusion of the extensions in a third party product.

14.1 Starting and Stopping `httpd`

During the installation process, a Bourne shell script named `httpd` was saved in `/etc/rc.d/init.d`. To manually stop and start your server, run `httpd` with either `stop` or `start` as an argument.

To start your server, type the command:

```
/etc/rc.d/init.d/httpd start
```

If you are running Apache as a secure server, you will be prompted to fill in your password. After you type it in, your server will start.

To stop your server, type the command:

```
/etc/rc.d/init.d/httpd stop
```

The command `restart` is a shorthand way of stopping and then starting your server. The `restart` command explicitly stops and then starts your server, so you will be prompted for your password if you are running Apache as a secure server. The `restart` command looks like the following:

```
/etc/rc.d/init.d/httpd restart
```

If you just finished editing something in your `httpd.conf` file, you do not need to explicitly stop and start your server. Instead, you may use the `reload` command. When you use `reload`, you will not need to type in your password (which is required if you are running Apache as a secure server). Your password will remain cached across reloads, but it will not be cached between stops and starts. The `reload` command looks like the following:

```
/etc/rc.d/init.d/httpd reload
```

By default, the `httpd` process will start automatically when your machine boots. If you are running Apache as a secure server, you will be prompted for the secure server's password after the machine boots, unless you generated a key for your secure server without password protection.

14.2 Configuration Directives in `httpd.conf`

The Apache Web server's configuration file is `/etc/httpd/conf/httpd.conf`. The `httpd.conf` file is well-commented and somewhat self-explanatory. Its default configuration will work for most people, so you probably will not need to change the directives in `httpd.conf`. However, you may want to be familiar with the most important configuration options.

The empty `srm.conf` and `access.conf` files are also in the `/etc/httpd/conf` directory. The `srm.conf` and `access.conf` files were formerly used, along with `httpd.conf`, as configuration files for Apache.

If you need to configure Apache, you simply edit `httpd.conf`, and then either reload, or stop and start the `httpd` process. How to reload, stop and start Apache is covered in Section 14.1, *Starting and Stopping `httpd`.*

Before you edit `httpd.conf`, you should first copy the original file to something like `httpd.conf-old` (or to any name you want). Then, if you make a mistake while you are editing the configuration file, you will have a backup.

If you do make a mistake, and your Web server does not work correctly, the first place to look is at what you just edited in `httpd.conf`. Make sure that you did not make a typo. The next place to look is your Web server's error log (`/var/log/httpd/error_log`). The error log may not be easy to interpret, depending on your level of experience. If you have just experienced a problem, however, the last entries in the error log should provide information about what has happened.

The next sections provide short descriptions of the directives which are included in `httpd.conf`, in the order that you will find them in the file. These descriptions are not exhaustive. If you need more information, please refer to the Apache documentation provided in HTML format at http://*your_domain*/manual/ or to the Apache group documentation at http://httpd.apache.org/docs/. For more information about mod_ssl directives, refer to the documentation included in HTML format as http://*your_domain*/manual/mod/mod_ssl/, or see the mod_ssl *User Manual* at http://www.modssl.org/docs/2.7/.

14.2.1 `ServerType`

Your `ServerType` can be either `inetd` or `standalone`. By default, your Web server is set to `ServerType standalone`.

`ServerType standalone` means that the server is started once and then that server handles all of the connections. `ServerType inetd` means that for every HTTP connection, a new instance of the server is started. Each server instance handles the connection and exits when the connection is ended. As you can probably imagine, using `inetd` is very inefficient. Another problem is that `inetd` may not work correctly, according to the Apache group. And finally, since Red Hat Linux 7.1 uses `xinetd`, additional configuration will be needed to get `xinetd` to start the server. For these reasons, you will want to leave your Web server's `ServerType` set to `standalone`.

14.2.2 `ServerRoot`

The `ServerRoot` is the top-level directory which will contain the server's files. Both your secure and non-secure servers are set to use a `ServerRoot` of `/etc/httpd`.

14.2.3 `LockFile`

`LockFile` sets the path to the lockfile used when the Apache server is compiled with either `USE_FCNTL_SERIALIZED_ACCEPT` or `USE_FLOCK_SERIALIZED_ACCEPT`. `LockFile` should normally be left at its default value.

14.2.4 `PidFile`

`PidFile` names the file in which the server records its process ID (pid). Your Web server is set to record its pid in `/var/run/httpd.pid`.

14.2.5 ScoreBoardFile

The `ScoreBoardFile` stores internal server process information, which is used for communication between the parent server process and its child processes. Your Web server's `ScoreBoardFile` is set to `/var/run/httpd.scoreboard`.

14.2.6 ResourceConfig

The `ResourceConfig` directive instructs the server to read the file named after `ResourceConfig` for more directives. The `ResourceConfig` directive is commented out, because your Web server only uses `httpd.conf` for configuration directives.

14.2.7 AccessConfig

The `AccessConfig` directive instructs the server to read the file named after `AccessConfig` for more directives, after it has read the file named by `ResourceConfig`. The `AccessConfig` directive is commented out, because your Web server only uses `httpd.conf` for configuration directives.

14.2.8 Timeout

`Timeout` defines, in seconds, the amount of time that your server will wait for receipts and transmissions during communications. Specifically, `Timeout` defines how long your server will wait to receive a GET request, how long it will wait to receive TCP packets on a POST or PUT request and how long it will wait between ACKs responding to TCP packets. `Timeout` is set to 300 seconds, which is appropriate for most situations.

14.2.9 KeepAlive

`KeepAlive` sets whether your server will allow persistent connections (in other words, more than one request per connection). `KeepAlive` can be used to prevent any one client from consuming too much of the server's resources. By default, `KeepAlive` is set to `on`, which means that your server allows persistent connections. You could set it to `off`, which would disable persistent connections. See Section 14.2.10, *`MaxKeepAliveRequests`* for a related way to limit requests per connection.

14.2.10 MaxKeepAliveRequests

This directive sets the maximum number of requests allowed per persistent connection. The Apache Group recommends a high setting, which will improve your server's performance. `MaxKeepAliveRequests` is set to `100` by default, which should be appropriate for most situations.

14.2.11 `KeepAliveTimeout`

`KeepAliveTimeout` sets the number of seconds your server will wait for a subsequent request, after a request has been served, before it closes the connection. Once a request has been received, the `Timeout` directive applies instead.

14.2.12 `MinSpareServers` and `MaxSpareServers`

The Apache Web server dynamically adapts to the perceived load by maintaining an appropriate number of spare server processes based on the traffic. The server checks the number of servers waiting for a request and kills some if there are more than `MaxSpareServers` or creates some if the number of servers is less than `MinSpareServers`.

Your server's default `MinSpareServers` is `5`; your server's default `MaxSpareServers` is `20`. These default settings should be appropriate in almost all situations. You should not increase the `MinSpareServers` to a very large number, since that will create a heavy processing load on your server even when traffic is light.

14.2.13 `StartServers`

`StartServers` sets how many server processes are created upon startup. Since your Web server dynamically kills and creates server processes based on traffic load, you will not ever need to change this parameter. Your Web server is set to start eight server processes at startup.

14.2.14 `MaxClients`

`MaxClients` sets a limit on the total number of server processes (i.e., simultaneously connected clients) that can run at one time. You want to keep `MaxClients` at a high number (your server's default is set to 150), because no one else will be allowed to connect once that number of simultaneously connected clients is reached. You can not set `MaxClients` to higher than 256 without recompiling Apache. The main reason for having `MaxClients` is so that a runaway Web server does not crash your operating system.

14.2.15 `MaxRequestsPerChild`

`MaxRequestsPerChild` sets the total number of requests each child server process serves before the child dies. The main reason for setting `MaxRequestsPerChild` is to avoid long-lived process induced memory leaks. The default `MaxRequestsPerChild` for your server is `100`.

14.2.16 `Listen`

The `Listen` command identifies the ports on which your Web server will accept incoming requests. Your Web server is set to listen to port 80 for non-secure Web communications and (in the virtual host tags that define the secure server) to port 443 for secure Web communications.

If you set Apache to listen to a port under 1024, you must be root to start it. For port 1024 and above, `httpd` can be started as a regular user.

`Listen` can also be used to specify particular IP addresses over which the server will accept connections.

14.2.17 BindAddress

`BindAddress` is a way of specifying which IP addresses your server will listen to. You should use the `Listen` directive instead if you need this functionality. `BindAddress` is not used by your Web server; by default it is commented out in `httpd.conf`.

14.2.18 LoadModule

`LoadModule` is used to load in Dynamic Shared Object (DSO) modules. More information on the Apache's DSO support, including exactly how to use the `LoadModule` directive, can be found in Section 14.3, *Adding Modules to Your Server*. Note that the order of the modules is important, so do not move them around.

14.2.19 IfDefine

The `<IfDefine>` and `</IfDefine>` tags surround configuration directives that are applied if the "test" stated in the `<IfDefine>` tag is true. The directives are ignored if the test is false.

The test in the `<IfDefine>` tags is a parameter name (for example, `HAVE_PERL`). If the parameter is defined, meaning that it is provided as an argument to the server's start-up command, then the test is true. In this case, when your Web server is started, the test is true and the directives contained in the `IfDefine` tags are applied.

By default, `<IfDefine HAVE_SSL>` tags surround the virtual host tags for your secure server. `<IfDefine HAVE_SSL>` tags also surround the `LoadModule` and `AddModule` directives for the `ssl_module`.

14.2.20 ClearModuleList

The `ClearModuleList` directive is located immediately before the long list of `AddModule` directives. `ClearModuleList` erases the server's built-in list of active modules. Then the list of `AddModule` directives re-creates the list, immediately after `ClearModuleList`.

14.2.21 AddModule

`AddModule` is the directive used to create a complete list of all available modules. You will use the `AddModule` directive if you add your own module in as a DSO. For more information on how `AddModule` is used for DSO support, see Section 14.3, *Adding Modules to Your Server*.

14.2.22 `ExtendedStatus`

The `ExtendedStatus` directives controls whether Apache generates basic (`off`) or detailed server status information (`on`), when the `server-status` handler is called. `Server-status` is called using `Location` tags. More information on calling `server-status` is included in Section 14.2.71, *`Location`*.

14.2.23 `Port`

Normally, `Port` defines the port that your server is listening to. Your Web server, however, is listening to more than one port by default, since the `Listen` directive is also being used. When `Listen` directives are in effect, your server listens at all of those ports. See the description of the `Listen` directive for more information about `Listen`.

The `Port` command is also used to specify the port number used to construct a canonical name for your server. See Section 14.2.39, *`UseCanonicalName`* for more information about your server's canonical name.

14.2.24 `User`

The `User` directive sets the userid used by the server to answer requests. `User`'s setting determines the server's access. Any files inaccessible to this user will also be inaccessible to your website's visitors. The default for `User` is `apache`.

The `User` should only have privileges so that it can access files which are supposed to be visible to the outside world. The `User` is also the owner of any CGI processes spawned by the server. The `User` should not be allowed to execute any code which is not intended to be in response to HTTP requests.

Note

Unless you know exactly what you are doing, do not set the `User` to root. Using root as the `User` will create large security holes for your Web server.

The parent `httpd` process first runs as root during normal operations but is then immediately handed off to the apache user. The server must start as root because it needs to bind to a port below 1024 (the default port for secure Web communications is port 443; the default port for non-secure Web communications is port 80). Ports below 1024 are reserved for system use, so they can not be used by anyone but root. Once the server has attached itself to its port, however, it hands the process off to the `User` before it accepts any connection requests.

14.2.25 Group

The Group directive is similar to the User. The Group sets the group under which the server will answer requests. The default Group is also apache.

14.2.26 ServerAdmin

ServerAdmin should be the email address of the Web server's administrator. This email address will show up in error messages on server-generated Web pages, so users can report a problem by sending email to the server administrator. ServerAdmin is set by default to root@localhost.

Typically, a good way to set up ServerAdmin is to set it to webmaster@your_domain.com. Then alias webmaster to the person responsible for the Web server in /etc/aliases. Finally, run /usr/bin/newaliases to add the new alias.

14.2.27 ServerName

You can use ServerName to set a hostname for your server which is different from your host's real name. For example, you might want to use www.your_domain.com when your server's real name is actually foo.your_domain.com. Note that the ServerName must be a valid Domain Name Service (DNS) name that you have the right to use (do not just make something up).

If you do specify a ServerName, be sure its IP address and server name pair are included in your /etc/hosts file.

14.2.28 DocumentRoot

The DocumentRoot is the directory which contains most of the HTML files which will be served in response to requests. The default DocumentRoot for both the non-secure and secure Web servers is /var/www/html. For example, the server might receive a request for the following document:

```
http://your_domain/foo.html
```

The server will look for the following file in the default directory:

```
/var/www/html/foo.html
```

If you want to change the DocumentRoot so that it is not shared by the secure and the non-secure Web servers, see Section 14.4, *Using Virtual Hosts*.

14.2.29 Directory

<Directory /path/to/directory> and </Directory> tags are used to enclose a group of configuration directives that are meant to apply only to that directory and all of its subdirectories. Any directive which is applicable to a directory may be used within <Directory> tags. <File> tags can be used in the same way, to apply to a specific file.

By default, very restrictive parameters are applied to the root directory, using the `Options` (see Section 14.2.30, *`Options`*) and `AllowOverride` (see Section 14.2.31, *`AllowOverride`*) directives. Under this configuration, any directory on your system which needs more permissive settings has to be explicitly given those settings.

Using `Directory` tags, the `DocumentRoot` is defined to have less rigid parameters, so that HTTP requests can be served from it.

The `cgi-bin` directory is set up to allow the execution of CGI scripts, with the `ExecCGI` option. If you need to execute a CGI script in another directory, you will need to set `ExecCGI` for that directory. For example, if your `cgi-bin` is `/var/www/cgi-bin`, but you want to execute CGI scripts from within `/home/my_cgi_directory`, add an `ExecCGI` directive to a set of `Directory` directives like the following to your `httpd.conf` file:

```
<Directory /home/my_cgi_directory>
    Options +ExecCGI
</Directory>
```

To allow CGI script execution in `/home/my_cgi_directory`, you will need to take a few extra steps besides setting `ExecCGI`. You will also need to have the `AddHandler` directive uncommented to identify files with the `.cgi` extension as CGI scripts. See Section 14.2.65, *`AddHandler`* for instructions on setting `AddHandler`. Permissions for CGI scripts, and the entire path to the scripts, must be set to 0755. Finally, the owner of the script and the owner of the directory must be the same user.

14.2.30 `Options`

The `Options` directive controls which server features are available in a particular directory. For example, under the restrictive parameters specified for the root directory, `Options` is set to only `FollowSymLinks`. No features are enabled, except that the server is allowed to follow symbolic links in the root directory.

By default, in your `DocumentRoot` directory, `Options` is set to include `Indexes`, `Includes` and `FollowSymLinks`. `Indexes` permits the server to generate a directory listing for a directory if no `DirectoryIndex` (for example, `index.html`) is specified. `Includes` means that server-side includes are permitted. `FollowSymLinks` allows the server to follow symbolic links in that directory.

You will also need to include `Options` statements for directories within virtual hosts directives, if you want your virtual hosts to recognize those `Options`.

For example, server side includes are already enabled inside the `/var/www/html` directory, because of the `Options Includes` line within the `<Directory "/var/www/html">` directives section. However, if you want a virtual host to recognize that server side includes are allowed within

/var/www/html, you will need to include a section like the following within your virtual host's tags:

```
<Directory /var/www/html>
Options Includes
</Directory>
```

14.2.31 AllowOverride

The AllowOverride directive sets whether or not any Options can be overridden by the declarations in an .htaccess file. By default, both the root directory and the DocumentRoot are set to allow no .htaccess overrides.

14.2.32 Order

The Order directive simply controls the order in which allow and deny directives are evaluated. Your server is configured to evaluate the Allow directives before the deny directives for your DocumentRoot directory.

14.2.33 Allow

Allow specifies which requester can access a given directory. The requester can be all, a domain name, an IP address, a partial IP address, a network/netmask pair, etc. Your DocumentRoot directory is configured to Allow requests from all (i.e., anyone).

14.2.34 Deny

Deny works just like Allow, but you are specifying who is denied access. Your DocumentRoot is not configured to Deny requests from anyone.

14.2.35 UserDir

UserDir is the name of the subdirectory within each user's home directory where they should place personal HTML files which are to be served by the Web server. By default, the subdirectory is public_html. For example, the server might receive the following request:

```
http://your_domain/~username/foo.html
```

The server would look for the file:

```
/home/username/public_html/foo.html
```

In the above example, /home/username is the user's home directory (note that the default path to users' home directories may be different on your system).

Make sure that the permissions on the users' home directories are set correctly. Users' home directories must be set to 0755. The read (r) and execute (x) bits must be set on the users' `public_html` directories (0755 will work). Files that will be served in users' `public_html` directories must be set to at least 0644.

14.2.36 DirectoryIndex

The `DirectoryIndex` is the default page served by the server when a user requests an index of a directory by specifying a forward slash (/) at the end of the directory name.

When a user requests the page http://*your_domain*/*this_directory*/, they are going to get either the `DirectoryIndex` page if it exists, or a server-generated directory list. The default for `DirectoryIndex` is `index.html index.htm index.shtml index.php index.php4 index.php3 index.cgi`. The server will try to find any one of these files, and will return the first one it finds. If it does not find any of these files and `Options Indexes` is set for that directory, the server will generate and return a listing, in HTML format, of the subdirectories and files in the directory.

14.2.37 AccessFileName

`AccessFileName` names the file which the server should use for access control information in each directory. By default, your Web server is set to use `.htaccess`, if it exists, for access control information in each directory.

Immediately after the `AccessFileName` directive, a set of `Files` tags apply access control to any file beginning with a `.ht`. These directives deny Web access to any `.htaccess` files (or other files which begin with `.ht`) for security reasons.

14.2.38 CacheNegotiatedDocs

By default, your Web server asks proxy servers not to cache any documents which were negotiated on the basis of content (that is, they may change over time or because of the input from the requester). If you uncomment `CacheNegotiatedDocs`, you are disabling that function and proxy servers will be allowed to cache the documents from then on.

14.2.39 UseCanonicalName

`UseCanonicalName` is set by default to `on`. `UseCanonicalName` allows the server to construct an URL that references itself, using `ServerName` and `Port`. When the server refers to itself in response to requests from clients, it uses this URL. If you set `UseCanonicalName` to `off`, the server will instead use the value that came in the request from the client to refer to itself.

14.2.40 TypesConfig

`TypesConfig` names the file which sets the default list of MIME type mappings (filename extensions to content types). The default `TypesConfig` file is `/etc/mime.types`. Instead of editing `/etc/mime.types`, the recommended way to add MIME type mappings is to use the `AddType` directive.

14.2.41 DefaultType

`DefaultType` sets a default content type for the Web server to use for documents whose MIME types can not be determined. Your Web server defaults to assume a plain text content type for any file with an indeterminate content type.

14.2.42 IfModule

`<IfModule>` and `</IfModule>` tags surround directives that are conditional. The directives contained within the `IfModule` tags are processed under one of two conditions. The directives are processed if the module contained within the starting `<IfModule>` tag is compiled in to the Apache server. Or, if an "!" (an exclamation point) is included before the module name, the directives are processed only if the module in the starting `<IfModule>` tag is *not* compiled in.

The `mod_mime_magic.c` file is included in these `IfModule` tags. The mod_mime_magic module can be compared to the UNIX `file` command, which looks at a few bytes of a file's contents, then uses "magic numbers" and other hints in order to figure out the MIME type of the file.

If the mod_mime_magic module is compiled in to Apache, these `IfModule` tags tell the mod_mime_magic module where the hints definition file is: `/usr/share/magic` in this case.

The mod_mime_magic module is not compiled in by default. If you would like to use it, see Section 14.3, *Adding Modules to Your Server*, for instructions on how to add modules to your server.

14.2.43 HostnameLookups

`HostnameLookups` can be set to `on`, `off` or `double`. If you allow `HostnameLookups` (by setting it to `on`), your server will automatically resolve the IP address for each connection which requests a document from your Web server. Resolving the IP address means that your server will make one or more connections to the DNS in order to find out the hostname that corresponds to a particular IP address. If you set `HostnameLookups` to `double`, your server will perform a double-reverse DNS. In other words, after a reverse lookup is performed, a forward lookup is performed on the result. At least one of the IP addresses in the forward lookup must match the address from the first reverse lookup.

Generally, you should leave HostnameLookups set to off, because the DNS requests add a load to your server and may slow it down. If your server is busy, the effects of HostnameLookups may be quite noticeable.

HostnameLookups are also an issue for the Internet as a whole. All of the individual connections made to look up each hostname add up. Therefore, for your own Web server's benefit, as well as for the good of the Internet as a whole, you should leave HostnameLookups set to off.

14.2.44 ErrorLog

ErrorLog names the file where server errors are logged. As this directive indicates, the error log file for your Web server is /var/log/httpd/error_log.

The error log is a good place to look if your Web server ever generates any errors or fails and you are not sure what happened.

14.2.45 LogLevel

LogLevel sets how verbose the error messages in the error logs will be. LogLevel can be set (from least verbose to most verbose) to emerg, alert, crit, error, warn, notice, info or debug. Your Web server's LogLevel is set to warn.

14.2.46 LogFormat

The LogFormat directives in your httpd.conf file set up a format for the messages in your access log. Hopefully, this format will make your access log more readable.

14.2.47 CustomLog

CustomLog identifies the log file and the log file format. In your Web server's default configuration, CustomLog defines the log file in which accesses to your Web server are recorded: /var/log/httpd/access_log. You will need to know the location of this file if you want to generate any access-based server performance statistics for your Web server.

CustomLog also sets the log file format to common. The common logfile format looks like this:

```
remotehost rfc931 authuser [date] "request" status bytes
```

remotehost

The remote hostname. If the hostname is not available from DNS, or if HostnameLookups is set to Off, then *remotehost* will be the IP address of the remote host.

rfc931

Not used. You will see a - in the log file in its place.

authuser

If authentication was required, this is the username with which the user identified him or herself. Usually, this is not used, so you will see a `-` in its place.

[date]

The date and time of the request.

"request"

The request string exactly as it came from the browser or client.

status

The HTTP status code which was returned to the browser or client.

bytes

The size of the document.

The `CustomLog` command can be used to set up specific log files to record referers (the URL for the Web page which linked to a page on your Web server) and/or agents (the browsers used to retrieve Web pages from your Web server). The relevant `CustomLog` lines are commented out, as shown, but you should uncomment them if you want those two log files:

```
#CustomLog /var/log/httpd/referer_log referer
#CustomLog /var/log/httpd/agent_log agent
```

Alternatively, you can also set the `CommonLog` directive to use a `combined` log by uncommenting the following line:

```
#CustomLog /var/log/httpd/access_log combined
```

A `combined` log will add the referer and agent fields to the end of the common log fields. If you want to use a `combined` log, you will need to comment out the `CustomLog` directive setting your access log to the common logfile format.

14.2.48 ServerSignature

The `ServerSignature` directive adds a line containing the Apache server version and the `ServerName` of the serving host to any server-generated documents (for example, error messages sent back to clients). `ServerSignature` is set to `on` by default. You can change it to `off`, so no signature line will be added, or you can change it to `EMail`. `EMail` will add a `mailto:ServerAdmin` HTML tag to the signature line.

14.2.49 `Alias`

The `Alias` setting allows directories to be outside the `DocumentRoot` directory and yet still accessible to the Web server. Any URL ending in the alias will automatically resolve to the alias' path. By default, one alias is already set up. An `icons` directory can be accessed by the Web server, but the directory is not in the `DocumentRoot`. The `icons` directory, an alias, is actually `/var/www/icons/`, not `/var/www/html/icons/`.

14.2.50 `ScriptAlias`

The `ScriptAlias` setting defines where CGI scripts (or other types of scripts) can be found. Generally, you do not want to leave CGI scripts within the `DocumentRoot`. If CGI scripts are in `DocumentRoot`, they could potentially be viewed as text documents. Even if you do not care if people can see (and then use) your CGI scripts, revealing how they work creates opportunities for unscrupulous people to exploit any security holes in the script, and may create a security risk for your server. By default, the `cgi-bin` directory is a `ScriptAlias` of `/cgi-bin/`, and is actually located in `/var/www/cgi-bin/`.

Your `/var/www/cgi-bin` directory has `Options ExecCGI` set, meaning that execution of CGI scripts is permitted within that directory.

See Section 14.2.65, *`AddHandler`* and Section 14.2.29, *`Directory`* for instructions on how to execute CGI scripts in directories other than the `cgi-bin`.

14.2.51 `Redirect`

When a Web page is moved, `Redirect` can be used to map the old URL to a new URL. The format is as follows:

```
Redirect /path/foo.html http://new_domain/path/foo.html
```

So, if an HTTP request is received for a page which used to be found at http://*your_domain*/*path*/foo.html, the server will send back the new URL (http://*new_domain*/*path*/foo.html) to the client, which should attempt to fetch the document from the new URL.

14.2.52 `IndexOptions`

`IndexOptions` controls the appearance of server generated directing listings, by adding icons and file descriptions, etc. If `Options Indexes` is set (see Section 14.2.30, *`Options`*), your Web server may generate a directory listening when your Web server receives an HTTP request like the following:

http://*your_domain*/*this_directory*/

First, your Web server looks in that directory for a file from the list after the DirectoryIndex directive (usually, index.html). If your Web server does not find one of those files, it creates an HTML directory listing of the subdirectories and files in the directory. You can modify the appearance of this directory listing using certain directives in httpd.conf, including IndexOptions.

Your default configuration sets FancyIndexing on. If FancyIndexing is turned on, clicking on the column headers in the directory listing will sort the order of the display by that header. Another click on the same header will switch from ascending to descending order and back. FancyIndexing also shows different icons for different files, depending upon file extensions. If you use the AddDescription directive and turn FancyIndexing on, then a short description of a file will be included in the server generated directory listing.

IndexOptions has a number of other parameters which can be set to control the appearance of server generated directories. Parameters include IconHeight and IconWidth, to make the server include HTML HEIGHT and WIDTH tags for the icons in server generated Web pages; IconsAreLinks, for making the icons act as part of the HTML link anchor along with the filename, and others.

14.2.53 AddIconByEncoding

This directive names icons which will be displayed by files with MIME encoding, in server generated directory listings. For example, by default, your Web server shows the compressed.gif icon next to MIME encoded x-compress and x-gzip files in server generated directory listings.

14.2.54 AddIconByType

This directive names icons which will be displayed next to files with MIME types in server generated directory listings. For example, your server is set to show the icon text.gif next to files with a mime-type of "text," in server generated directory listings.

14.2.55 AddIcon

AddIcon tells the server which icon to show in server generated directory listings for certain file types or for files with certain extensions. For example, your Web server is set to show the icon binary.gif for files with .bin or .exe extensions.

14.2.56 DefaultIcon

DefaultIcon names the icon to show in server generated directory listings for files which have no other icon specified. The unknown.gif image file is the DefaultIcon for those files by default.

14.2.57 AddDescription

You can use AddDescription to show text that you specify for certain files, in server generated directory listings (you will also need to enable FancyIndexing as an IndexOptions). You can

name specific files, wildcard expressions or file extensions to specify the files which this directive should apply to. For example, you could use the following line:

```
AddDescription "A file that ends in .ni" .ni
```

In server generated directory listings, all files with extensions of `.ni` would have the description `A file that ends in .ni` after the filename. Note that you will also need `FancyIndexing` turned on.

14.2.58 ReadmeName

`ReadmeName` names the file which (if it exists in the directory) will be appended to the end of server generated directory listings. The Web server will first try to include the file as an HTML document and then try to include it as plaintext. By default, `ReadmeName` is set to `README`.

14.2.59 HeaderName

`HeaderName` names the file which (if it exists in the directory) will be prepended to the start of server generated directory listings. Like `ReadmeName`, the server will try to include it as an HTML document if possible, or in plaintext if not.

14.2.60 IndexIgnore

`IndexIgnore` lists file extensions, partial filenames, wildcard expressions or full filenames. The Web server will not include any files which match any of those parameters in server generated directory listings.

14.2.61 AddEncoding

`AddEncoding` names filename extensions which should specify a particular encoding type. `AddEncoding` can also be used to instruct some browsers (not all) to uncompress certain files as they are downloaded.

14.2.62 AddLanguage

`AddLanguage` associates filename extensions with specific content languages. This directive is mostly useful for content negotiation, when the server returns one of several documents based on the client's language preference as set in their browser.

14.2.63 LanguagePriority

`LanguagePriority` allows you to set precedence for different languages in which to serve files, which will be in effect if the client expressed no preference for language in their browser.

14.2.64 AddType

Use the AddType directive to define MIME type and file extension pairs. For example, if you are using PHP4, your Web server is using the AddType directive to make your Web server recognize files with PHP extensions (.php4, .php3 .phtml .php) as PHP MIME types.

The following AddType line tells your server to recognize the .shtml file extension (for server side includes):

```
AddType text/html .shtml
```

You will need to include the above line within the virtual host tags for any virtual hosts which should allow server side includes.

14.2.65 AddHandler

AddHandler maps file extensions to specific handlers. For example, the cgi-script handler can be used matched with the extension .cgi to automatically treat a file ending with .cgi as a CGI script. This will work, even for files outside of the ScriptAlias directory, as long as you follow the instructions provided here.

You have a CGI AddHandler line in your httpd.conf file:

```
AddHandler cgi-script .cgi
```

You will have to uncomment the line. Then Apache will execute CGI scripts for files ending in .cgi, even if they are outside of the ScriptAlias, which is set by default to locate your /cgi-bin/ directory in /var/www/cgi-bin/.

You will also need to set ExecCGI as an Options for any directory containing a CGI script. See Section 14.2.29, *Directory* for more information about setting ExecCGI for a directory. Additionally, you will need to make sure the permissions are set correctly for the CGI scripts and the directories containing CGI scripts. CGI scripts and the entire directory path to the scripts must be set to 0755. Finally, the owner of the directory and the owner of the script file must be the same user.

You will need to add the same AddHandler line to your VirtualHost setup, if you are using virtual hosts and you want them to also recognize CGI scripts outside the ScriptAlias.

In addition to CGI scripts, your Web server also uses AddHandler to process server-parsed HTML and imagemap files.

14.2.66 Action

Action allows you to specify a MIME content type and CGI script pair, so that whenever a file of that media type is requested, a particular CGI script will be executed.

14.2.67 MetaDir

MetaDir specifies the name of a directory where your Web server should look for files containing meta information (extra HTTP headers) to include when serving documents.

14.2.68 MetaSuffix

MetaSuffix specifies the filename suffix for the file that contains meta information (extra HTTP headers), which should be located in the MetaDir directory.

14.2.69 ErrorDocument

By default, in the event of a problem or error, your Web server outputs a simple (and usually cryptic) error message back to the requesting client. Instead of using the default, you can use ErrorDocument to configure your Web server so that it outputs a customized message or redirects the client to a local or external URL. The ErrorDocument directive simply associates a HTTP response code with a message or a URL which will be sent back to the client.

14.2.70 BrowserMatch

The BrowserMatch directive allows your server to define environment variables and/or take appropriate actions based on the User-Agent HTTP header field, which identifies the client's browser. By default, your Web server uses BrowserMatch to deny connections to specific browsers with known problems and also to disable keepalives and HTTP header flushes for browsers that are known to have problems with those actions.

14.2.71 Location

<Location> and </Location> tags allow you to specify access control based on the URL.

The next use of Location tags is located within IfModule mod_perl.c tags. These configuration directives are in effect if the mod_perl.so DSO is loaded. See Section 14.3, *Adding Modules to Your Server* for more information about adding modules to Apache.

The Location tags name the /var/www/perl directory (an Alias for /perl) as the directory from which Perl scripts will be served. If a document is requested with an URL containing /perl in the path, your Web server will look in /var/www/perl/ for the appropriate Perl script.

Several other <Location> options are commented out in your httpd.conf file. If you want to enable the functionality they provide, you will need to uncomment the appropriate section of directives.

Immediately after the Perl directives discussed previously, your httpd.conf file includes a section of directives for enabling HTTP PUT (used by Netscape Gold's publish feature, which can post Web

pages to a Web server). If you want to allow HTTP PUT, you will need to uncomment this entire section:

```
#Alias /upload /tmp
#<Location /upload>
#    EnablePut On
#    AuthType Basic
#    AuthName Temporary
#    AuthUserFile /etc/httpd/conf/passwd
#    EnableDelete Off
#    umask 007
#    <Limit PUT>
#       require valid-user
#    </Limit>
#</Location>
```

You will also need to uncomment the following lines at the beginning of `httpd.conf` so that the mod_put module is loaded in to Apache:

```
#LoadModule put_module          modules/mod_put.so
#AddModule mod_put.c
```

If you want to allow people connecting from your domain to see server status reports, you should uncomment the next section of directives:

```
#<Location /server-status>
#    SetHandler server-status
#    Order deny,allow
#    Deny from all
#    Allow from .your_domain.com
#</Location>
```

You must replace `.your_domain.com` with your second level domain name.

If you want to provide server configuration reports (including installed modules and configuration directives) to requests from inside your domain, you will need to uncomment the following lines:

```
#<Location /server-info>
#    SetHandler server-info
#    Order deny,allow
#    Deny from all
#    Allow from .your_domain.com
#</Location>
```

Again, you must fill in `.your_domain.com`.

The next section of directives use `Location` tags to allow access to the documentation in `/usr/share/doc` (for example, with a URL like http://*your_domain*/doc/whatever.html). These directives only allow this access to requests made from the localhost.

Another use of the `Location` tags is a commented-out section which is intended to track attacks on your Web server which exploit an old bug from pre-Apache 1.1 days. If you want to track these requests, uncomment the following lines:

```
#<Location /cgi-bin/phf*>
#    Deny from all
#    ErrorDocument 403 http://phf.apache.org/phf_abuse_log.cgi
#</Location>
```

If these lines are uncommented, your Web server will redirect any requests which end in `/cgi-bin/phf*` to a logging CGI script run by the Apache Group.

14.2.72 `ProxyRequests`

If you uncomment the `IfModule` tags surrounding the `ProxyRequests` directives, your Apache server will also function as a proxy server. You will also need to load the `mod_proxy` module. For instructions on how to load in modules, see Section 14.3, *Adding Modules to Your Server*.

14.2.73 `ProxyVia`

The `ProxyVia` command controls whether or not an HTTP Via: header line is sent along with requests or replies which go through the Apache proxy server. The Via: header will show the hostname if `ProxyVia` is set to `On`, the hostname and Apache version for `Full`, any Via: lines will be passed along unchanged for `Off`, and Via: lines will be removed for `Block`.

14.2.74 Cache Directives

A number of cache directives are commented out in the proxy `IfModule` tags mentioned above. If you are using the proxy server functionality and you want to also enable the proxy cache, you should uncomment the cache directives as described. The default settings for your cache directives should be appropriate for most configurations.

`CacheRoot` sets the name of the directory which will contain cached files. The default `CacheRoot` is `/var/cache/httpd`.

`CacheSize` sets how much space the cache can use, in KB. The default `CacheSize` is `5` KB.

`CacheGcInterval` sets a number of hours. After that number of hours, files in the cache will be deleted if the cache is using more space than allowed by `CacheSize`. The default for `CacheGcInterval` is four hours.

Cached HTML documents will be retained (without a reload from the originating Web server) in the cache for a maximum number of hours set by `CacheMaxExpire`. The default is `24` hours.

The `CacheLastModifiedFactor` affects the creation of an expiry (expiration) date for a document which did not come from its originating server with its own expiry set. The default `CacheLastModifiedFactor` is set to `0.1`, meaning that the expiry date for such documents equals one-tenth of the amount of time since the document was last modified.

`CacheDefaultExpire` is the expiry time in hours for a document that was received using a protocol that does not support expiry times. The default is set to one hour.

Any document that is retrieved from a host and/or domain that matches one set in `NoCache` will not be cached. If you know of hosts or domains from which you do not want to cache documents, uncomment `NoCache` and set their domains or hostnames here.

14.2.75 `NameVirtualHost`

You will need to use the `NameVirtualHost` directive for the IP address (and port number if necessary) of any name-based virtual hosts you are setting up. The name-based virtual hosts configuration is used when you want to set up different virtual hosts for different domains, but you do not have (or do not want to use) different IP addresses for all of the different domain names for which your Web server serves documents.

Note

You cannot use name-based virtual hosts with your secure server. Any name-based virtual hosts you set up will only work with non-secure HTTP connections and not with SSL connections.

You cannot use name-based virtual hosts with your secure server because the SSL handshake (when the browser accepts the secure Web server's authenticating certificate) occurs before the HTTP request which identifies the correct name-based virtual host. In other words, authentication occurs before there is any identification of different name-based virtual hosts. If you want to use virtual hosts with your secure server, you will need to use IP address-based virtual hosts.

If you are using name-based virtual hosts, uncomment the `NameVirtualHost` configuration directive and add the correct IP address for your server after `NameVirtualHost`. Then add more information about the different domains using the `Virtual Host` tags which surround the `ServerName` for each virtual host, plus any other configuration directives which are only applicable to that virtual host.

14.2.76 `VirtualHost`

`<VirtualHost>` and `</VirtualHost>` tags surround any configuration directives which are intended to apply to a virtual host. Most configuration directives can be used within virtual host tags, and then they only apply to that particular virtual host.

A set of commented out `VirtualHost` tags surround some example configuration directives and placeholders for the information you would need to fill in to set up a virtual host. Please see Section 14.4, *Using Virtual Hosts*, for more information about virtual hosts.

14.2.77 `SetEnvIf`

The Apache configuration directive `SetEnvIf` is used to disable HTTP keepalive and to allow SSL to close the connection without a close notify alert from the client browser. This setting is necessary for certain browsers that do not reliably shut down the SSL connection.

14.2.78 SSL Configuration Directives

The SSL directives in your server's `httpd.conf` file are included to enable secure Web communications using SSL and TLS.

For more information on SSL directives, please point your browser to http://*your_domain*/manual/mod/mod_ssl/. More information on SSL directives is also available at http://www.modssl.org/docs/2.7/ssl_reference.html, a chapter in a Web document about mod_ssl by Ralf Engelschall. The same document, the mod_ssl *User Manual*, begins at http://www.modssl.org/docs/2.7/ and is a great reference source for mod_ssl and for Web cryptography in general. This manual provides general information about securing your Web server in Chapter 13, *Using Apache as a Secure Web Server*.

Note

Do not modify your SSL directives unless you are absolutely sure about what you are doing. In most cases, the SSL directives are configured appropriately as installed.

14.3 Adding Modules to Your Server

Since Apache 1.3 supports DSOs, you can easily load Apache modules or compile in your own modules to your Web server. DSO support means that modules may be loaded at runtime. Since the modules are only loaded as necessary, they will not use any memory unless they are loaded and less memory will be needed overall.

The Apache Group provides complete DSO Documentation at http://httpd.apache.org/docs/dso.html. After installation of your server, you can also check http://*your_domain*/manual/mod/ for documentation on Apache modules in HTML format (if you installed the `apache-manual` package). A short description of how to load modules is provided next. If you need more details, check the URLs provided.

For Apache to use a dynamically shared module, that module must have a `LoadModule` line and an `AddModule` line in `httpd.conf`. By default, many modules have these two lines already included in `httpd.conf`, but a few of the less commonly used modules are commented out. The commented out modules were included during compilation, but they are not loaded by default.

If you need to use one of those non-loaded modules, look in the `httpd.conf` file to see all the available modules. Each of the available modules has a corresponding `LoadModule` line. To show you an example, the `LoadModule` section begins with these seven lines:

```
#LoadModule mmap_static_module modules/mod_mmap_static.so
LoadModule vhost_alias_module modules/mod_vhost_alias.so
LoadModule env_module         modules/mod_env.so
LoadModule config_log_module  modules/mod_log_config.so
LoadModule agent_log_module   modules/mod_log_agent.so
LoadModule referer_log_module modules/mod_log_referer.so
#LoadModule mime_magic_module  modules/mod_mime_magic.so
```

Most of the lines are not commented out, indicating that each associated module was compiled in and is loaded in by default. The first line is commented out, which means that the corresponding module (`mmap_static_module`) was compiled in but not loaded.

To make Apache load an unloaded module, first uncomment the corresponding `LoadModule` line. For example, if you wanted to make Apache load in the `mime_magic_module`, uncomment this line:

```
#LoadModule mime_magic_module modules/mod_mime_magic.so
```

Next, you need to uncomment the corresponding line from the `AddModule` section in `httpd.conf`. To continue with our previous example, uncomment the `mod_mime_magic` line, which looks like the following:

```
#AddModule mod_mime_magic.c
```

Once you have uncommented the `LoadModule` and `AddModule` lines for the module that you want to load in, stop and start Apache, as covered in Section 14.1, *Starting and Stopping `httpd`*. After starting, the module should be loaded in to Apache.

If you have your own module, you can add it to the `httpd.conf` file so that it is compiled in and loaded as a DSO. If you want to do this, you need to install the `apache-devel` package, as covered in Chapter 13, *Using Apache as a Secure Web Server*. You need the `apache-devel` package because

it installs the include files, the header files and the APache eXtenSion (APXS) support tool. APXS uses the include files and the header files to compile your module so that it will work with Apache.

WARNING

If you plan to use the Apache Configuration Tool, a GUI utility provided with Red Hat Linux, you must not compile your own modules to your Apache Web server or edit your Apache Web server's `httpd.conf` configuration file. Conversely, if you want to add modules to Apache or edit `httpd.conf` by hand, do not use the Apache Configuration Tool.

If you need more information on the Apache Configuration Tool, please see the *Official Red Hat Linux Customization Guide*.

If you have written your own module or are borrowing someone else's, you should be able to use APXS to compile your module sources outside the Apache source tree, without needing to tweak any compiler and/or linker flags. If you need more information on APXS, please see the Apache documentation at http://httpd.apache.org/docs/dso.html.

Once you have compiled your module using APXS, put your module into `/usr/lib/apache`. Then your module needs both a `LoadModule` line and an `AddModule` line in the `httpd.conf` file, just as described previously for Apache's own modules. After the `LoadModule` list in `httpd.conf`, add a line for the shared object file for your module like the following:

```
LoadModule foo_module  modules/mod_foo.so
```

Note that you will need to change the name of the module and the name of your shared object file as appropriate.

At the end of the `AddModule` list in `httpd.conf`, add a line for the source code file for your module like the following:

```
AddModule mod_foo.c
```

Note that you will need to change the name of the source code file as appropriate.

Once you have completed the previous steps, stop and start your Web server as outlined in Section 14.1, *Starting and Stopping `httpd`*. If you have done everything correctly and your module is correctly coded, your Web server should find your module and load it in as it starts.

14.3.1 The mod_ssl Security Module

The mod_ssl security portion of the Web server is provided as a Dynamic Shared Object (DSO). This means that the Apache Web server can be re-compiled by users if the EAPI extension patch from the mod_ssl security module is applied to Apache. Follow the instructions for building mod_ssl into Apache included with the mod_ssl documentation, but add the following flag:

```
--with-eapi-only
```

The complete command line should look like the following:

```
./configure [userflags] --with-eapi-only
```

Then build and install Apache.

Note

Red Hat cannot support re-compiled versions of the Apache Web server. Installation of the shipped version is supported, but if you re-compile Apache, you are on your own. Please do not re-compile Apache unless you know exactly what you are doing.

14.4 Using Virtual Hosts

WARNING

If you plan to use the Apache Configuration Tool, a GUI utility provided with Red Hat Linux, you may not edit your Apache Web server's `httpd.conf` configuration file. Conversely, if you want to edit `httpd.conf` by hand, do not use the Apache Configuration Tool.

If you need more information on the Apache Configuration Tool, please see the *Official Red Hat Linux Customization Guide*.

You can use Apache's virtual hosts capability to run different servers for different IP addresses, different host names or different ports on the same machine. If you are interested in using virtual hosts, complete information is provided in the Apache documentation on your machine or on the Web at http://httpd.apache.org/docs/vhosts/.

Note

You cannot use name-based virtual hosts with your secure Web server, because the SSL handshake (when the browser accepts the secure Web server's certificate) occurs before the HTTP request which identifies the appropriate name-based virtual host. If you want to use name-based virtual hosts, they will only work with your non-secure Web server.

Virtual hosts are configured within the `httpd.conf` file, as described in Section 14.2, *Configuration Directives in `httpd.conf`*. Please review that section before you start to change the virtual hosts configuration on your machine.

14.4.1 The Secure Web Server Virtual Host

The default configuration of your Web server runs a non-secure and a secure server. Both servers use the same IP address and host name, but they listen on different ports, and the secure server is a virtual host. This configuration enables you to serve both secure and non-secure documents in the most efficient manner possible. As you may know, secure HTTP transmissions take more time than non-secure, because a lot more information is being passed back and forth during secure transactions. So using your secure server for non-secure Web traffic is not a good idea.

The configuration directives for your secure server are contained within virtual host tags in the `httpd.conf` file. If you need to change something about the configuration of your secure server, you will need to change the configuration directives inside virtual host tags in the `httpd.conf` file. If you want to enable certain features (for example, server side includes) for your secure server, they will need to be enabled within the virtual host tags that define your secure server.

The non-secure Web server is configured as the "non-virtual" host in the `httpd.conf` file. In other words, the non-secure Web server's configuration options are outside of the virtual host tags in `httpd.conf`. If you want to change something about your non-secure Web server, you will need to change the configuration directives in `httpd.conf` outside of the virtual host tags.

By default, both the secure and the non-secure Web servers share the same `DocumentRoot`, a configuration directive specified in `httpd.conf`. In other words, the secure and the non-secure Web server look in the same place for the HTML files that they provide in response to requests. By default, the `DocumentRoot` is set to `/var/www/html`.

To change the `DocumentRoot` so that it is no longer shared by both the secure server and the non-secure server, change one of the `DocumentRoot` directives in `httpd.conf`. The `DocumentRoot` outside the virtual host tags defines the `DocumentRoot` for your non-secure Web server. The `DocumentRoot` within the virtual host tags that define your secure server is for your secure server.

If for some reason you want to disable the non-secure Web server on your machine, you can. Your secure server listens on port 443, the default port for secure Web communications, while your non-secure Web server listens on port 80, the default port for non-secure Web communications. To stop the non-secure Web server from accepting connections, in `httpd.conf`, find the line which reads:

```
Port 80
```

Change the above line so that it reads:

```
Port 443
```

Then comment out the `Listen 80` line.

After these two steps, your Web server will be accepting connections on port 443, the default port for secure Web communications. However, your server will not accept connections on port 80, the default port for non-secure communications, so the non-secure Web server will be effectively disabled.

14.4.2 Setting Up Virtual Hosts

Most people will probably use their Web server as it is configured. Therefore, they will be using the built-in virtual hosts capability, but they will not have to do any manipulation of the virtual hosts directives in `httpd.conf`. However, if you would like to use the virtual hosts capability for some other reason, you can.

To create a virtual host, you will need to alter the virtual host lines, provided as an example, in `httpd.conf`, or create your own virtual host section. (Remember that name-based virtual hosts will not work with your secure server — you will need to use IP address-based virtual hosts if you need SSL-enabled virtual hosts. Your non-secure server, however, will support both IP address and name-based virtual hosts.)

The virtual host example lines read as follows:

```
#<VirtualHost ip.address.of.host.some_domain.com>
#    ServerAdmin webmaster@host.some_domain.com
#    DocumentRoot /www/docs/host.some_domain.com
#    ServerName host.some_domain.com
#    ErrorLog logs/host.some_domain.com-error_log
#    CustomLog logs/host.some_domain.com-access_log common
#</VirtualHost>
```

Uncomment all of the lines. Then add the correct information for your machine and/or your virtual host to each line.

In the first line, change `ip.address.of.host.some_domain.com` to your server's IP address. Change the `ServerName` to a *valid* DNS name to use for the virtual host. (In other words, do not just make something up. Ask your system administrator if you do not know how to get a valid domain name.)

You will also need to uncomment one of the `NameVirtualHost` lines in `httpd.conf`:

```
#NameVirtualHost 12.34.56.78:80
#NameVirtualHost 12.34.56.78
```

Uncomment one of the lines and change the IP address to the IP address (and port if necessary) for that virtual host.

Many other configuration directives can be placed between the virtual host tags, depending upon why you are setting up a virtual host.

If you set up a virtual host and want it to listen on a non-default port (80 is the default port for non-secure Web communications; 443 is the default port for secure Web communications), you will need to set up a virtual host for that port and add a `Listen` directive to `httpd.conf`, corresponding to that port.

To have a virtual host work specifically for that port, add the port number to the first line of the virtual host configuration. The first line should look something like the following:

```
<VirtualHost ip_address_of_your_server:12331>
```

This line would create a virtual host that listens on port 12331. Substitute the port number you want to use for `12331` in the previous example.

Underneath the `Listen` lines in `httpd.conf`, add a line like the following, which will instruct your Web server to listen on port 12331:

```
Listen 12331
```

You must restart `httpd` to start a new virtual host. See Section 14.1, *Starting and Stopping `httpd`* for instructions on how to start and stop `httpd`.

Much more complete information about creating and configuring both name-based and IP address-based virtual hosts is provided on the Web at http://httpd.apache.org/docs/vhosts/. Please check the Apache Group's virtual host documentation for more details on using virtual hosts.

Part IV Appendixes

A General Parameters and Modules

This appendix is provided to illustrate *some* of the possible parameters that may be needed by certain drivers[1] for particular hardware devices. In most cases, these additional parameters are unnecessary, since the kernel may already be able to use the device without them. You should only use the settings provided in this appendix if you are having trouble getting Red Hat Linux to use a particular device or you need to override the system's default parameters for the device.

During the installation of Red Hat Linux, some limits are placed on filesystems and particular device drivers supported by the kernel. After installation, however, support exists for all filesystems available under Linux. At the time of installation, the modularized kernel has support for (E)IDE devices (including ATAPI CD-ROM drives), SCSI adapters, and network cards.

Note

Because Red Hat Linux supports installation on many different types of hardware, some drivers (including those for SCSI adapters, network cards, and many CD-ROMs) are not built into the Linux kernel used by the installation program. Rather, they are available as modules and are loaded as you need them during the installation process. If necessary, you will have the chance to specify options for these modules when they are loaded from the driver disk.

To specify module parameters when a driver is loaded, type **`linux expert`** at the `boot:` prompt and insert the driver disk when prompted to do so by the installation program. After reading the driver disk, the installation program will ask you to select the type of device you are configuring. On that screen, you can elect to specify a module parameter. Then, the installation program will display a screen where you can type the correct parameters based on the particular type of device you are configuring.

After the installation is complete, you may want to rebuild a kernel that includes support for your specific hardware configuration. Note that in most cases, a custom-built kernel is not necessary. See the *Official Red Hat Linux Customization Guide* for more information about rebuilding your kernel.

1 A **driver** is a type of software that helps your system use a particular hardware device. Without the driver, the kernel may not know how to correctly utilize the device.

A.1 Specifying Module Parameters

If you are providing parameters upon loading a module, you can usually specify them using one of two different methods:

- Specify a full set of parameters in one statement. For example, the parameter `cdu31=0x340,0` could be used with a Sony CDU 31 or 33 at port 340 with no IRQ.
- Specify the parameters individually. This method is used when one or more parameters in the first set are not needed. For example, `cdu31_port=0x340 cdu31a_irq=0` can be used as the parameter for the same CD-ROM used as an example for the first method. An *OK* is used in the CD-ROM, SCSI, and Ethernet tables in this appendix to show where the first parameter method stops and the second method begins.

Note

Only use one method, and not both, when loading a module with particular parameters.

When a parameter has commas, make sure you do *not* put a space after a comma.

A.2 CD-ROM Module Parameters

Note

Not all of the CD-ROM drives that are listed are supported. Please check the Hardware Compatibility List on Red Hat's website at http://hardware.redhat.com to make sure your CD-ROM drive is supported.

Even though parameters are specified after loading the driver disk and specifying the device, one of the more commonly used parameters (`hd`*X*`=cdrom`) *can* be entered at the boot prompt (`boot:`) during installation. This exception to the rule is allowed since it deals with support for IDE/ATAPI CD-ROMs, which is already part of the kernel.

In the following tables, most modules listed without any parameters can either be auto-probed to find the hardware or they require you to manually change settings in the module source code and recompile.

Table A–1 Hardware Parameters

Hardware	Module	Parameters
ATAPI/IDE CD-ROM Drives		hd*X*=cdrom
Aztech CD268-01A, Orchid CD-3110, Okano/Wearnes CDD110, Conrad TXC, CyCDROM CR520, CyCDROM CR540 (non-IDE)	`aztcd.o`	aztcd=*io_port*
Sony CDU-31A CD-ROM	`cdu31a.o`	cdu31a=*io_port,IRQ OR* cdu31a_port=*base_addr* cdu31a_irq=*irq*
Philips/LMS CDROM drive 206 with cm260 host adapter card	`cm206.o`	cm206=*io_port,IRQ*
Goldstar R420 CD-ROM	`gscd.o`	gscd=*io_port*
ISP16, MAD16, or Mozart sound card CD-ROM interface (OPTi 82C928 and OPTi 82C929) with Sanyo/Panasonic, Sony, or Mitsumi drives	`isp16.o`	isp16=*io_port,IRQ,dma, drive_type OR* isp16_cdrom_base=*io_port* isp16_cdrom_irq=*IRQ* isp16_cdrom_dma=*dma* isp16_cdrom_type=*drive_type*
Mitsumi CD-ROM, Standard	`mcd.o`	mcd=*io_port,IRQ*
Mitsumi CD-ROM, Experimental	`mcdx.o`	mcdx=*io_port_1,IRQ_1, io_port_n,IRQ_n*
Optics storage 8000 AT "Dolphin" drive, Lasermate CR328A	`optcd.o`	
Parallel-Port IDE CD-ROM	`pcd.o`	
SB Pro 16 Compatible	`sbpcd.o`	sbpcd=*io_port*

Hardware	Module	Parameters
Sanyo CDR-H94A	`sjcd.o`	sjcd=*io_port OR* sjcd_base=*io_port*
Sony CDU-535 & 531 (some Procomm drives)	`sonycd535.o`	sonycd535=*io_port*

Here are some examples of these modules in use:

Table A–2 Hardware Parameters Configuration Examples

Configuration	Example
ATAPI CD-ROM, jumpered as master on the second IDE channel	hdc=cdrom
non-IDE Mitsumi CD-ROM on port 340, IRQ 11	mcd=0x340,11
Three non-IDE Mitsumi CD-ROM drives using the experimental driver, io ports 300, 304, and 320 with IRQs 5, 10 and 11	mcdx=0x300,5,0x304,10,0x320,11
Sony CDU 31 or 33 at port 340, no IRQ	cdu31=0x340,0 *OR* cdu31_port=0x340 cdu31a_irq=0
Aztech CD-ROM at port 220	aztcd=0x220
Panasonic-type CD-ROM on a SoundBlaster interface at port 230	sbpcd=0x230,1
Phillips/LMS cm206 and cm260 at IO 340 and IRQ 11	cm206=0x340,11
Goldstar R420 at IO 300	gscd=0x300
Mitsumi drive on a MAD16 soundcard at IO Addr 330 and IRQ 1, probing DMA	isp16=0x330,11,0,Mitsumi
Sony CDU 531 at IO address 320	sonycd535=0x320

Note

Most newer Sound Blaster cards come with IDE interfaces. For these cards, you do not need to use `sbpcd` parameters; only use `hd`*X* parameters.

A.3 SCSI parameters

Table A–3 SCSI Parameters

Hardware	Module	Parameters
3ware Storage Controller	`3w-xxxx.o`	
NCR53c810/820/720, NCR53c700/710/700-66	`53c7,8xx.o`	
AM53/79C974 (PC-SCSI) Driver	`AM53C974.o`	
Most Buslogic (now Mylex) cards with "BT" part number	`BusLogic.o`	BusLogic_Options=*option*,*option*,…
Mylex DAC960 RAID Controller	`DAC960.o`	
MCR53c406a-based SCSI	`NCR53c406a.o`	
Initio INI-9100UW	`a100u2w.o`	a100u2w=*io*,*IRQ*,*scsi_id*
Adaptec AACRAID	`aacraid.o`	
Advansys SCSI Cards	`advansys.o`	
Adaptec AHA-152x	`aha152x.o`	aha152x=*io*,*IRQ*,*scsi_id*
Adaptec AHA 154x amd 631x-based	`aha1542.o`	
Adaptec AHA 1740	`aha1740.o`	

Hardware	Module	Parameters
Adaptec AHA-274x, AHA-284x, AHA-29xx, AHA-394x, AHA-398x, AHA-274x, AHA-274xT, AHA-2842, AHA-2910B, AHA-2920C, AHA-2930/U/U2, AHA-2940/W/U/UW/AU/, U2W/U2/U2B/, U2BOEM, AHA-2944D/WD/UD/UWD, AHA-2950U2/W/B, AHA-3940/U/W/UW/, AUW/U2W/U2B, AHA-3950U2D, AHA-3985/U/W/UW, AIC-777x, AIC-785x, AIC-786x, AIC-787x, AIC-788x, AIC-789x, AIC-3860	`aic7xxx.o`	aic7xxx=*string*
ACARD ATP870U PCI SCSI Controller	`atp870u.o`	
Compaq Smart Array 5300 Controller	`cciss.o`	
Compaq Smart/2 RAID Controller	`cpqarray.o`	
Compaq FibreChannel Controller	`cpqfc.o`	
Domex DMX3191D	`dmx3191d.o`	
Data Technology Corp DTC3180/3280	`dtc.o`	

Hardware	Module	Parameters
DTP SCSI host adapters (EATA/DMA) PM2011B/9X ISA, PM2021A/9X ISA, PM2012A, PM2012B, PM2022A/9X EISA, PM2122A/9X, PM2322A/9X, SmartRAID PM3021, PM3222, PM3224	`eata.o`	eata=*port0*,*port1*,*port2*,… *options OR* eata io_port=*port0*,*port1*,*port2*,… option=*value*
DTP SCSI Adapters PM2011, PM2021, PM2041, PM3021, PM2012B, PM2022, PM2122, PM2322, PM2042, PM3122, PM3222, PM3332, PM2024, PM2124, PM2044, PM2144, PM3224, PM3334	`eata_dma.o`	
DTP EATA-PIO boards	`eata_pio.o`	
Sun Enterprise Network Array (FC-AL)	`fcal.o`	
Future Domain TMC-16*xx* SCSI	`fdomain.o`	
NCR5380 (generic driver)	`g_NCR5380.o`	
ICP RAID Controller	`gdth.o`	
I2O Block Driver	`i2o_block.o`	
IOMEGA MatchMaker parallel port SCSI adapter	`imm.o`	
Always IN2000 ISA SCSI card	`in2000.o`	in2000=*setup_string*:*value OR* in2000 *setup_string*=*value*
Initio INI-9X00U/UW SCSI host adapters	`initio.o`	
IBM ServeRAID	`ips.o`	
AMI MegaRAID 418, 428, 438, 466, 762	`megaraid.o`	

Hardware	Module	Parameters
NCR SCSI controllers with 810/810A/815/ 825/825A/860/875/876/895 chipsets	`ncr53c8xx.o`	ncr53c8xx=*option1*:*value1*,*option2*:*value2*,… *OR* ncr53c8xx="*option1*:*value1 option2*:*value2*…"
Pro Audio Spectrum/Studio 16	`pas16.o`	
PCI-2000 IntelliCache	`pci2000.o`	
PCI-2220I EIDE RAID	`pci2220i.o`	
SparcSTORAGE Array	`pluto.o`	
IOMEGA PPA3 parallel port SCSI host adapter	`ppa.o`	
Perceptive Solutions PSI-240I EIDE	`psi240i.o`	
Qlogic 1280	`qla1280.o`	
Qlogic 2*x*00	`qla2x00.o`	
QLogic Fast SCSI FASXXX ISA/VLB/PCMCIA	`qlogicfas.o`	
QLogic ISP2100 SCSI-FCP	`qlogicfc.o`	
QLogic ISP1020 Intelligent SCSI cards IQ-PCI, IQ-PCI-10, IQ-PCI-D	`qlogicisp.o`	
Qlogic ISP1020 SCSI SBUS	`qlogicpti.o`	
Seagate ST-01/02, Future Domain TMC-8*xx*	`seagate.o`	
Future Domain TMC-885, TMC-950	`seagate.o`	controller_type=2 base_address=*base_addr* irq=*IRQ*
Cards with the sym53c416 chipset	`sym53c416.o`	sym53c416=*PORTBASE*,*[IRQ]* *OR* sym53c416 io=*PORTBASE* irq=*IRQ*

Hardware	Module	Parameters
Trantor T128/T128F/T228 SCSI Host Adapter	t128.o	
Tekram DC-390(T) PCI	tmscsim.o	
UltraStor 14F/34F (not 24F)	u14-34f.o	
UltraStor 14F, 24F, and 34F	ultrastor.o	
WD7000 Series	wd7000.o	

Here are some examples of these modules in use:

Table A–4 SCSI Parameters Configuration Examples

Configuration	Example
Adaptec AHA1522 at port 330, IRQ 11, SCSI ID 7	aha152x=0x330,11,7
Adaptec AHA1542 at port 330	bases=0x330
Future Domain TMC-800 at CA000, IRQ 10	controller_type=2 base_address=0xca000 irq=10

A.4 Ethernet parameters

Table A–5 Ethernet Module Parameters

Hardware	Module	Parameters
3Com 3c501	3c501.o	3c501=*io_port,IRQ*
3Com 3c503 and 3c503/16	3c503.o	3c503=*io_port,IRQ OR* 3c503 io=*io_port_1,io_port_n* irq=*IRQ_1,IRQ_n*
3Com EtherLink Plus (3c505)	3c505.o	3c505=*io_port,IRQ OR* 3c505 io=*io_port_1,io_port_n* irq=*IRQ_1,IRQ_2*
3Com EtherLink 16	3c507.o	3c507=*io_port,IRQ OR* 3c507 io=*io_port* irq=*IRQ*
3Com EtherLink III	3c509.o	3c509=*IRQ*

Hardware	Module	Parameters
3Com ISA EtherLink XL "Corkscrew"	`3c515.o`	
3Com EtherLink PCI III/XL Vortex (3c590, 3c592, 3c595, 3c597) Boomerang (3c900, 3c905, 3c595)	`3c59x.o`	
RTL8139, SMC EZ Card Fast Ethernet	`8139too.o`	
Apricot 82596	`82596.o`	
Ansel Communications Model 3200	`ac3200.o`	ac3200=*io_port,IRQ OR* ac3200 io=*io_port_1,io_port_n* irq=*IRQ_1,IRQ_n*
Alteon AceNIC Gigabit	`acenic.o`	
Aironet Arlan 655	`arlan.o`	
Aironet 4500 PCI-ASI-i365 wireless	`aironet4500_card.o`	
Allied Telesis AT1700	`at1700.o`	at1700=*io_port,IRQ OR* at1700 io=*io_port* irq=*IRQ*
Tangent ATB-II, Novel NL-10000, Daystar Digital LT-200, Dayna DL2000, DaynaTalk PC (HL), COPS LT-95, Farallon PhoneNET PC II, III	`cops.o`	cops=*io_port,IRQ OR* cops io=*io_port* irq=*IRQ*
Modular driver for the COSA or SRP synchronous serial card	`cosa.o`	cosa=*io_port,IRQ,dma*
Crystal Semiconduc-torCS89[02]0	`cs89x0.o`	

Hardware	Module	Parameters
EtherWORKS DE425 TP/COAX EISA, DE434 TP PCI, DE435/450 TP/COAX/AUI PCI DE500 10/100 PCI Kingston, LinkSys, SMC8432, SMC9332, Znyx31[45], and Znyx346 10/100 cards with DC21040 (no SROM), DC21041[A], DC21140[A], DC21142, DC21143 chipsets	`de4x5.o`	de4x5=*io_port OR* de4x5 io=*io_port* de4x5 args='eth*X[fdx]* autosense=*MEDIA_STRING*'
D-Link DE-600 Ethernet Pocket Adapter	`de600.o`	
D-Link DE-620 Ethernet Pocket Adapter	`de620.o`	
DIGITAL DEPCA & EtherWORKS DEPCA, DE100, DE101, DE200 Turbo, DE201Turbo DE202 Turbo TP/BNC, DE210, DE422 EISA	`depca.o`	depca=*io_port,IRQ OR* depca io=*io_port* irq=*IRQ*
Digi Intl. RightSwitch SE-X EISA and PCI	`dgrs.o`	
Davicom DM9102(A)/DM9132/ DM9801 Fast Ethernet	`dmfe.o`	
Intel EtherExpress/1000 Gigabit	`e1000.o`	
Cabletron E2100	`e2100.o`	e2100=*io_port,IRQ,mem OR* e2100 io=*io_port* irq=*IRQ* mem=*mem*

Hardware	Module	Parameters
Intel EtherExpress Pro10	`eepro.o`	eepro=*io_port,IRQ OR* eepro io=*io_port* irq=*IRQ*
Intel i82557/i82558 PCI EtherExpressPro driver	`eepro100.o`	
Intel EtherExpress 16 (i82586)	`eexpress.o`	eexpress=*io_port,IRQ OR* eexpress io=*io_port* irq=*IRQ*
SMC EtherPower II 9432 PCI (83c170/175 EPIC series)	`epic100.o`	
Racal-Interlan ES3210 EISA	`es3210.o`	
ICL EtherTeam 16i/32 EISA	`eth16i.o`	eth16i=*io_port,IRQ OR* eth16i ioaddr=*io_port* IRQ=*IRQ*
EtherWORKS 3 (DE203, DE204 and DE205)	`ewrk3.o`	ewrk=*io_port,IRQ OR* ewrk io=*io_port* irq=*IRQ*
Fujitsu FMV-181/182/183/184	`fmv18x.o`	fmv18x=*io_port,IRQ OR* fmv18x io=*io_port* irq=*IRQ*
A Packet Engines GNIC-II Gigabit	`hamachi.o`	
Modular driver for the Comtrol Hostess SV11	`hostess_sv11.o`	hostess_sv11=*io_port,IRQ, DMABIT OR* hostess_sv11 io=*io_port* irq=*IRQ* dma=*DMABIT*
HP PCLAN/plus	`hp-plus.o`	hp-plus=*io_port,IRQ OR* hp-plus io=*io_port* irq=*IRQ*
HP LAN Ethernet	`hp.o`	hp=*io_port,IRQ OR* hp io=*io_port* irq=*IRQ*

Hardware	Module	Parameters
100VG-AnyLan Network Adapters HP J2585B, J2585A, J2970, J2973, J2573 Compex ReadyLink ENET100-VG4, FreedomLine 100/VG	`hp100.o`	hp100=*io_port,name OR* hp100 hp100_port=*io_port* hp100_name=*name*
IBM Token Ring 16/4	`ibmtr.o`	ibmtr=*io_port,IRQ,mem OR* ibmtr io=*io_port* irq=*IRQ* mem=*mem*
AT1500, HP J2405A, most NE2100/clone	`lance.o`	
Mylex LNE390 EISA	`lne390.o`	
	`ltpc.o`	ltpc=*io_port,IRQ OR* ltpc io=*io_port* irq=*IRQ*
MyriCOM MyriNET SBUS	`myri_sbus.o`	
NatSemi DP83815 Fast Ethernet	`natsemi.o`	
NE1000 / NE2000 (non-pci)	`ne.o`	ne=*io_port,IRQ OR* ne io=*io_port* irq=*IRQ*
PCI NE2000 cards RealTEk RTL-8029, Winbond 89C940, Compex RL2000, KTI ET32P2, NetVin, NV5000SC, Via 82C926, SureCom NE34	`ne2k-pci.o`	
Novell NE3210 EISA	`ne3210.o`	
MiCom-Interlan NI5010	`ni5010.o`	
NI5210 card (i82586 Ethernet chip)	`ni52.o`	ni52=*io_port,IRQ OR* ni52 io=*io_port* irq=*IRQ*
NI6510 Ethernet	`ni65.o`	

Hardware	Module	Parameters
Older DEC 21040, most 21*40 Ethernet	`old_tulip.o`	old_tulip=*io_port OR* old_tulip io=*io_port*
AMD PCnet32 and AMD PCnetPCI	`pcnet32.o`	
RedCreek Communications PCI	`rcpci.o`	
RealTek cards using RTL8129 or RTL8139 Fast Ethernet chipsets	`rtl8139.o`	
Sangoma S502/S508 multi-protocol FR	`sdla.o`	
Sangoma S502A, ES502A, S502E, S503, S507, S508, S509	`sdladrv.o`	
SysKonnect SK-98*XX* Gigabit	`sk98lin.o`	
SysKonnect Token Ring ISA/PCI Adapter, TR4/16(+) ISA or PCI, TR4/16 PCI, and older SK NET TR4/16 ISA cards	`sktr.o`	sktr=*io_port*,*IRQ*,*mem OR* sktr io=*io_port* irq=*IRQ* mem=*mem*
SMC Ultra and SMC EtherEZ ISA ethercard (8K, 83c790)	`smc-ultra.o`	smc-ultra=*io_port*,*IRQ OR* smc-ultra io=*io_port* irq=*IRQ*
SMC Ultra32 EISA Ethernet card (32K)	`smc-ultra32.o`	
SMC 9000 series of Ethernet cards	`smc9194.o`	smc9194=*io_port*,*IRQ OR* smc9194 io=*io_port* irq=*IRQ* ifport=*[0,1,2]*
Sun BigMac Ethernet	`sunbmac.o`	
Sundance ST201 Alta	`sundance.o`	

Hardware	Module	Parameters
Sun Happy Meal Ethernet	`sunhme.o`	
Sun Quad Ethernet	`sunqe.o`	
ThunderLAN	`tlan.o`	
Digital 21x4x Tulip PCI Ethernet cards SMC EtherPower 10 PCI(8432T/8432BT) SMC EtherPower 10/100 PCI(9332DST) DEC EtherWorks 100/10 PCI(DE500-XA) DEC EtherWorks 10 PCI(DE450) DEC QSILVER's, Znyx 312 etherarray Allied Telesis LA100PCI-T Danpex EN-9400, Cogent EM110	`tulip.o`	
VIA Rhine PCI Fast Ethernet cards with either the VIA VT86c100A Rhine-II PCI or 3043 Rhine-I D-Link DFE-930-TX PCI 10/100	`via-rhine.o`	
AT&T GIS (nee NCR) WaveLan ISA Card	`wavelan.o`	wavelan=*[IRQ,0],io_port,NWID*
WD8003 and WD8013-compatible Ethernet cards	`wd.o`	wd=*io_port,IRQ,mem, mem_end OR* wd io=*io_port* irq=*IRQ* mem=*mem* mem_end=*end*
Compex RL100ATX-PCI	`winbond.o`	
Packet Engines Yellowfin	`yellowfin.o`	
Z8530 based HDLC cards for AX.25	`z85230.o`	

Here are some examples of these modules in use:

Table A–6 Ethernet Parameter Configuration Examples

Configuration	Example
NE2000 ISA card at IO address 300 and IRQ 11	ne=0x300,11 ether=0x300,11,eth0
Wavelan card at IO 390, autoprobe for IRQ, and use the NWID to 0x4321	wavelan=0,0x390,0x4321 ether=0,0x390,0x4321,eth0

A.4.1 Using Multiple Ethernet Cards

You can use multiple Ethernet cards in one machine. If each card uses a different driver (for example, a 3c509 and a DE425), you simply need to add `alias` (and possibly `options`) lines for each card to `/etc/modules.conf`. See the *Official Red Hat Linux Customization Guide* for more information.

If any two Ethernet cards use the same driver (such as two 3c509 cards or a 3c595 and a 3c905), you will need to either give the two card addresses on the driver's options line (for ISA cards) or simply add one `alias` line for each card (for PCI cards).

For additional information about using more than one Ethernet card, see the *Linux Ethernet-HOWTO* at http://www.redhat.com/mirrors/LDP/HOWTO/Ethernet-HOWTO.html.

B An Introduction to Disk Partitions

Disk partitions are a standard part of the personal computer landscape and have been for quite some time. However, with many people purchasing computers featuring preinstalled operating systems, relatively few people understand how partitions work. This chapter attempts to explain the reasons for and use of disk partitions so your Red Hat Linux installation will be as simple and painless as possible.

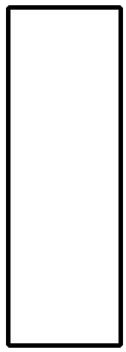
If you are reasonably comfortable with disk partitions, you could skip ahead to Section B.1.4, *Making Room For Red Hat Linux* for more information on the process of freeing up disk space to prepare for a Red Hat Linux installation. This section also discusses the partition naming scheme used by Linux systems, sharing disk space with other operating systems, and related topics.

B.1 Hard Disk Basic Concepts

Hard disks perform a very simple function — they store data and reliably retrieve it on command.

When discussing issues such as disk partitioning, it's important to know a bit about the underlying hardware. Unfortunately, it's easy to become bogged down in details. Therefore, let's use a simplified diagram of a disk drive to help explain what is really happening when a disk drive is partitioned. Figure B–1, *An Unused Disk Drive* shows a brand-new, unused disk drive.

Figure B–1 An Unused Disk Drive

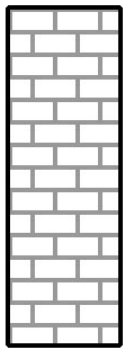


Not much to look at, is it? But if we are talking about disk drives on a basic level, it will do. Let's say that we'd like to store some data on this drive. As things stand now, it won't work. There's something we need to do first…

B.1.1 It's Not What You Write, it's How You Write It

Experienced computer users probably got this one on the first try. We need to **format** the drive. Formatting (usually known as "making a **filesystem**") writes information to the drive, creating order out of the empty space in an unformatted drive.

Figure B–2 Disk Drive with a Filesystem



As Figure B–2, *Disk Drive with a Filesystem* implies, the order imposed by a filesystem involves some trade-offs:

- A small percentage of the drive's available space is used to store filesystem-related data and can be considered as overhead.
- A filesystem splits the remaining space into small, consistently-sized segments. For Linux, these segments are known as **blocks**. [1]

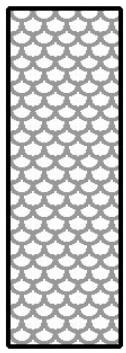
Given that filesystems make things like directories and files possible, these tradeoffs are usually seen as a small price to pay.

It's also worth noting that there is no single, universal filesystem. As Figure B–3, *Disk Drive with a Different Filesystem* shows, a disk drive may have one of many different filesystems written on it. As you might guess, different filesystems tend to be incompatible; that is, an operating system that supports one filesystem (or a handful of related filesystem types) may not support another. This last statement is not a hard-and-fast rule, however. For example, Red Hat Linux supports a wide variety

[1] Blocks really *are* consistently sized, unlike our illustrations. Keep in mind, also, that an average disk drive contains thousands of blocks. But for the purposes of this discussion, please ignore these minor discrepancies.

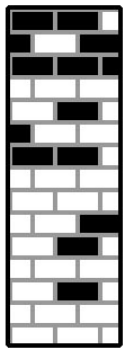
of filesystems (including many commonly used by other operating systems), making data interchange between different filesystems easy.

Figure B–3 Disk Drive with a Different Filesystem



Of course, writing a filesystem to disk is only the beginning. The goal of this process is to actually *store* and *retrieve* data. Let's take a look at our drive after some files have been written to it.

Figure B–4 Disk Drive with Data Written to It



As Figure B–4, *Disk Drive with Data Written to It* shows, 14 of the previously-empty blocks are now holding data. However, by simply looking at this picture, we cannot determine exactly how many files reside on this drive. There may be as few as one or as many as 14 files, as all files use at least one block and some files use multiple blocks. Another important point to note is that the used blocks do not have to form a contiguous region; used and unused blocks may be interspersed. This is known as **fragmentation**. Fragmentation can play a part when attempting to resize an existing partition.

As with most computer-related technologies, disk drives changed over time after their introduction. In particular, they got bigger. Not larger in physical size, but bigger in their capacity to store information. And, this additional capacity drove a fundamental change in the way disk drives were used.

B.1.2 Partitions: Turning One Drive Into Many

As disk drive capacities soared, some people began to wonder if having all of that formatted space in one big chunk was such a great idea. This line of thinking was driven by several issues, some philosophical, some technical. On the philosophical side, above a certain size, it seemed that the additional space provided by a larger drive created more clutter. On the technical side, some filesystems were never designed to support anything above a certain capacity. Or the filesystems *could* support larger

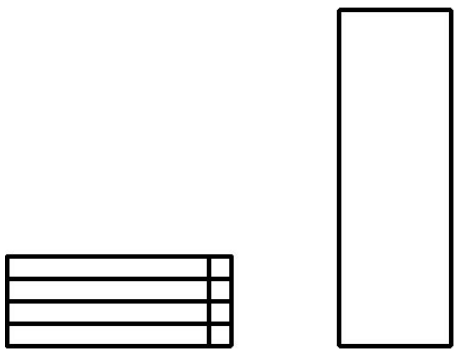
drives with a greater capacity, but the overhead imposed by the filesystem to track files became excessive.

The solution to this problem was to divide disks into **partitions**. Each partition can be accessed as if it was a separate disk. This is done through the addition of a **partition table**.

Note

While the diagrams in this chapter show the partition table as being separate from the actual disk drive, this is not entirely accurate. In reality, the partition table is stored at the very start of the disk, before any filesystem or user data. But for clarity, we'll keep it separate in our diagrams.

Figure B–5 Disk Drive with Partition Table



As Figure B–5, *Disk Drive with Partition Table* shows, the partition table is divided into four sections. Each section can hold the information necessary to define a single partition, meaning that the partition table can define no more than four partitions.

Each partition table entry contains several important characteristics of the partition:

- The points on the disk where the partition starts and ends

- Whether the partition is "active"
- The partition's type

Let's take a closer look at each of these characteristics. The starting and ending points actually define the partition's size and location on the disk. The "active" flag is used by some operating systems' boot loaders. In other words, the operating system in the partition that is marked "active" will be booted.

The partition's type can be a bit confusing. The type is a number that identifies the partition's anticipated usage. If that statement sounds a bit vague, that's because the meaning of the partition type is a bit vague. Some operating systems use the partition type to denote a specific filesystem type, to flag the partition as being associated with a particular operating system, to indicate that the partition contains a bootable operating system, or some combination of the three.

Table B–1, *Partition Types* contains a listing of some popular (and obscure) partition types, along with their numeric values.

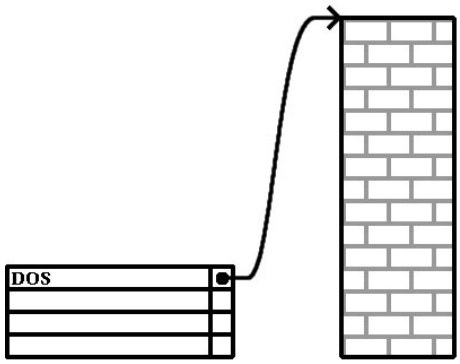
Table B–1 Partition Types

Partition Type	Value	Partition Type	Value
Empty	00	Novell Netware 386	65
DOS 12-bit FAT	01	PIC/IX	75
XENIX root	02	Old MINIX	80
XENIX usr	03	Linux/MINUX	81
DOS 16-bit <=32M	04	Linux swap	82
Extended	05	Linux native	83
DOS 16-bit >=32	06	Linux extended	85
OS/2 HPFS	07	Amoeba	93
AIX	08	Amoeba BBT	94
AIX bootable	09	BSD/386	a5
OS/2 Boot Manager	0a	OpenBSD	a6
Win95 FAT32	0b	NEXTSTEP	a7
Win95 FAT32 (LBA)	0c	BSDI fs	b7
Win95 FAT16 (LBA)	0e	BSDI swap	b8
Win95 Extended (LBA)	0f	Syrinx	c7

Partition Type	Value	Partition Type	Value
Venix 80286	40	CP/M	db
Novell	51	DOS access	e1
Microport	52	DOS R/O	e3
GNU HURD	63	DOS secondary	f2
Novell Netware 286	64	BBT	ff

By this point, you might be wondering how all this additional complexity is normally used. See Figure B–6, *Disk Drive With Single Partition* for an example.

Figure B–6 Disk Drive With Single Partition



In many cases, there is only a single partition spanning the entire disk, essentially duplicating the method used before partitions. The partition table has only one entry used, and it points to the start of the partition.

We have labeled this partition as being of the "DOS" type. Although it is only one of several possible partition types listed in Table B–1, *Partition Types*, it is adequate for the purposes of this discussion.

This is a typical partition layout for most newly purchased computers with a consumer version of Microsoft Windows™ preinstalled.

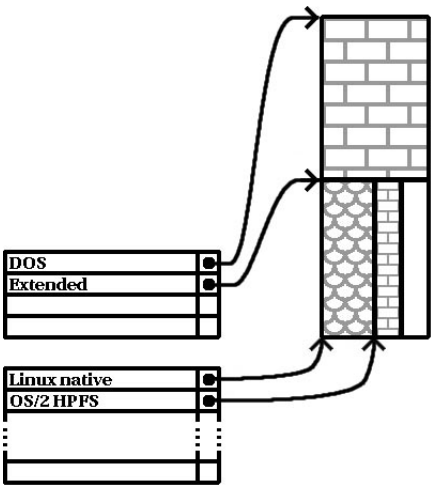
B.1.3 Partitions within Partitions – An Overview of Extended Partitions

Of course, over time it became obvious that four partitions would not be enough. As disk drives continued to grow, it became more and more likely that a person could configure four reasonably-sized partitions and still have disk space left over. There needed to be some way of creating more partitions.

Enter the extended partition. As you may have noticed in Table B–1, *Partition Types*, there is an "Extended" partition type. It is this partition type that is at the heart of extended partitions.

When a partition is created and its type is set to "Extended," an extended partition table is created. In essence, the extended partition is like a disk drive in its own right —— it has a partition table that points to one or more partitions (now called **logical partitions**, as opposed to the four **primary partitions**) contained entirely within the extended partition itself. Figure B–7, *Disk Drive With Extended Partition* shows a disk drive with one primary partition and one extended partition containing two logical partitions (along with some unpartitioned free space).

Figure B–7 Disk Drive With Extended Partition



As this figure implies, there is a difference between primary and logical partitions -- there can only be four primary partitions, but there is no fixed limit to the number of logical partitions that can exist. (However, in reality, it is probably not a good idea to try to define and use more than 12 logical partitions on a single disk drive.)

Now that we have discussed partitions in general, let's see how to use this knowledge to install Red Hat Linux.

B.1.4 Making Room For Red Hat Linux

There are three possible scenarios you may face when attempting to repartition your hard disk:

- Unpartitioned free space is available
- An unused partition is available
- Free space in an actively used partition is available

Let's look at each scenario in order.

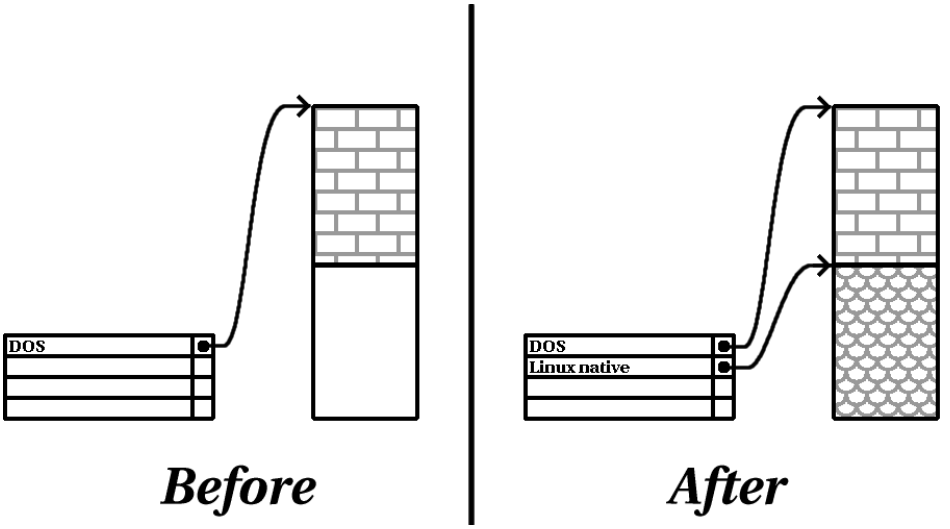
Note

Please keep in mind that the following illustrations are simplified in the interest of clarity and do not reflect the exact partition layout that you will encounter when actually installing Red Hat Linux.

Using Unpartitioned Free Space

In this situation, the partitions already defined do not span the entire hard disk, leaving unallocated space that is not part of any defined partition. Figure B–8, *Disk Drive with Unpartitioned Free Space* shows what this might look like.

Figure B–8 Disk Drive with Unpartitioned Free Space



If you think about it, an unused hard disk also falls into this category. The only difference is that *all* the space is not part of any defined partition.

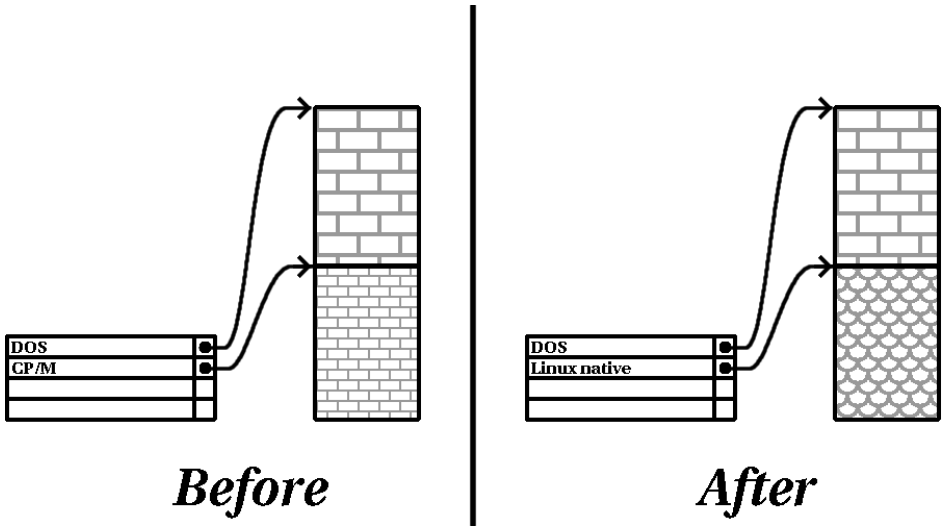
In any case, you can simply create the necessary partitions from the unused space. Unfortunately, this scenario, although very simple, is not very likely (unless you've just purchased a new disk just for Red Hat Linux). Most pre-installed operating systems are configured to take up all available space on a disk drive (see *Using Free Space from an Active Partition* in Section B.1.4).

Let's move on to a slightly more common situation.

Using Space from an Unused Partition

In this case, maybe you have one or more partitions that you do not use any longer. Perhaps you've dabbled with another operating system in the past, and the partition(s) you dedicated to it never seem to be used anymore. Figure B–9, *Disk Drive With an Unused Partition* illustrates such a situation.

Figure B–9 Disk Drive With an Unused Partition



If you find yourself in this situation, you can use the space allocated to the unused partition. You will first need to delete the partition, and then create the appropriate Linux partition(s) in its place. You can either delete the partition using the DOS `fdisk` command, or you will be given the opportunity to do so during a custom-class installation.

Using Free Space from an Active Partition

This is the most common situation. It is also, unfortunately, the hardest to handle. The main problem is that, even if you have enough free space, it's presently allocated to a partition that is already in use. If you purchased a computer with pre-installed software, the hard disk most likely has one massive partition holding the operating system and data.

Aside from adding a new hard drive to your system, you have two choices:

Destructive Repartitioning

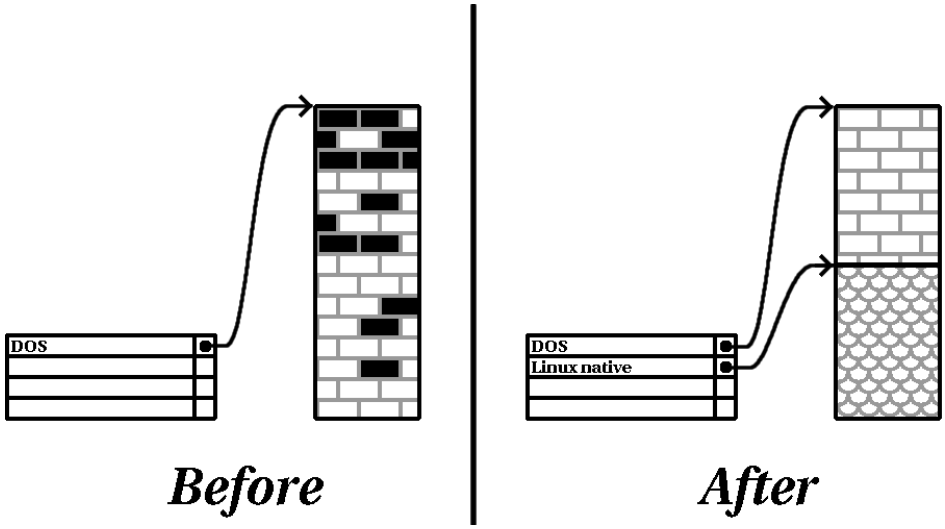
> Basically, you delete the single large partition and create several smaller ones. As you might imagine, any data you had in the original partition is destroyed. This means that making a complete backup is necessary. For your own sake, make two backups, use verification (if available in your backup software), and try to read data from your backup *before* you delete the partition.

CAUTION

If there was an operating system of some type installed on that partition, it will need to be reinstalled as well. Be aware that some computers sold with pre-installed operating systems may not include the CD-ROM media to reinstall the original operating system. The best time to notice if this applies to your system is *before* you destroy your original partition and its operating system installation.

After creating a smaller partition for your existing software, you can reinstall any software, restore your data, and continue your Red Hat Linux installation. Figure B–10, *Disk Drive Being Destructively Repartitioned* shows this being done.

Figure B–10 Disk Drive Being Destructively Repartitioned



As Figure B–10, *Disk Drive Being Destructively Repartitioned* shows, any data present in the original partition will be lost without proper backup!

Non-Destructive Repartitioning

Here, you run a program that does the seemingly impossible: it makes a big partition smaller without losing any of the files stored in that partition. Many people have found this method to be reliable and trouble-free. What software should you use to perform this feat? There are several disk management software products on the market. You will have to do some research to find the one that is best for your situation.

While the process of non-destructive repartitioning is rather straightforward, there are a number of steps involved:

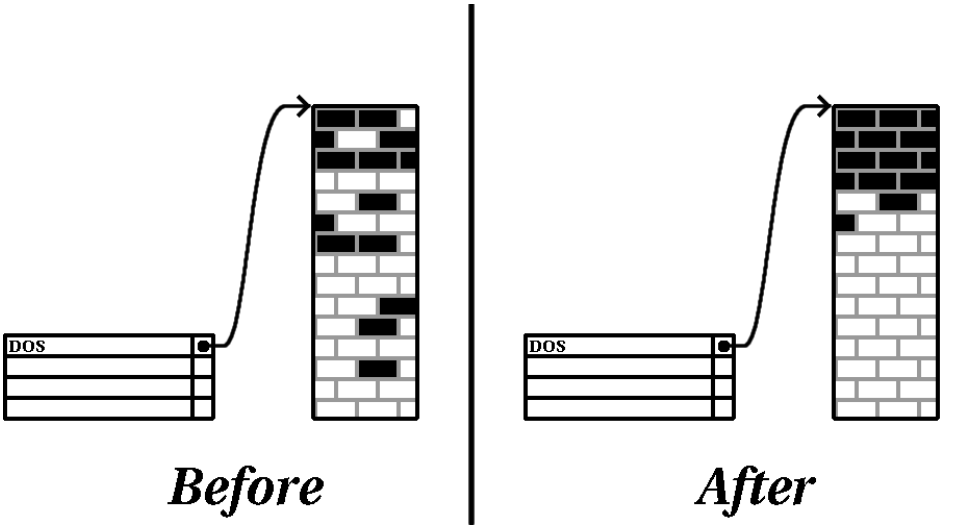
- Compress existing data
- Resize the existing partition
- Create new partition(s)

Let’s take a look at each step in a bit more detail.

Compress existing data

As Figure B–11, *Disk Drive Being Compressed* shows, the first step is to compress the data in your existing partition. The reason for doing this is to rearrange the data such that it maximizes the available free space at the "end" of the partition.

Figure B–11 Disk Drive Being Compressed

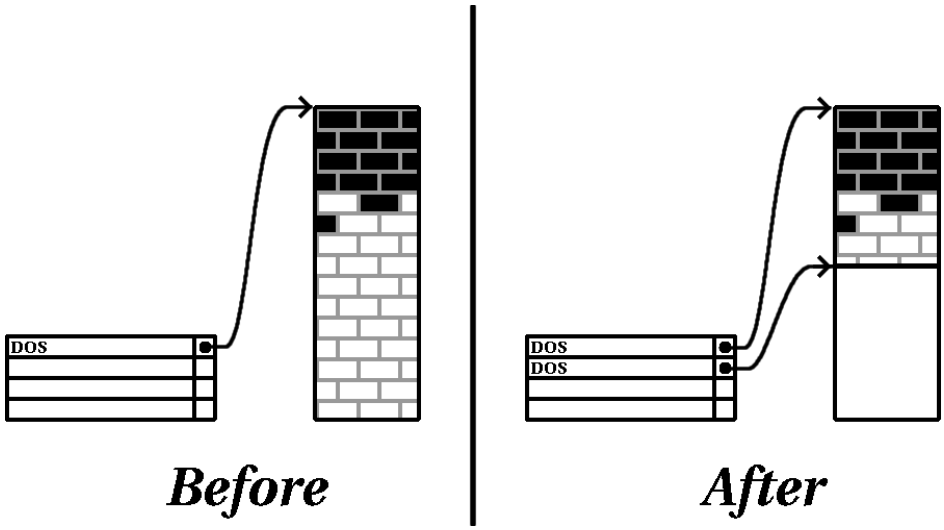


This step is crucial. Without it, the location of your data could prevent the partition from being resized to the extent desired. Note also that, for one reason or another, some data cannot be moved. If this is the case (and it severely restricts the size of your new partition(s)), you may be forced to destructively repartition your disk.

Resize the existing partition

Figure B–12, *Disk Drive with Partition Resized* shows the actual resizing process. While the actual result of the resizing operation varies depending on the software used, in most cases the newly freed space is used to create an unformatted partition of the same type as the original partition.

Figure B–12 Disk Drive with Partition Resized

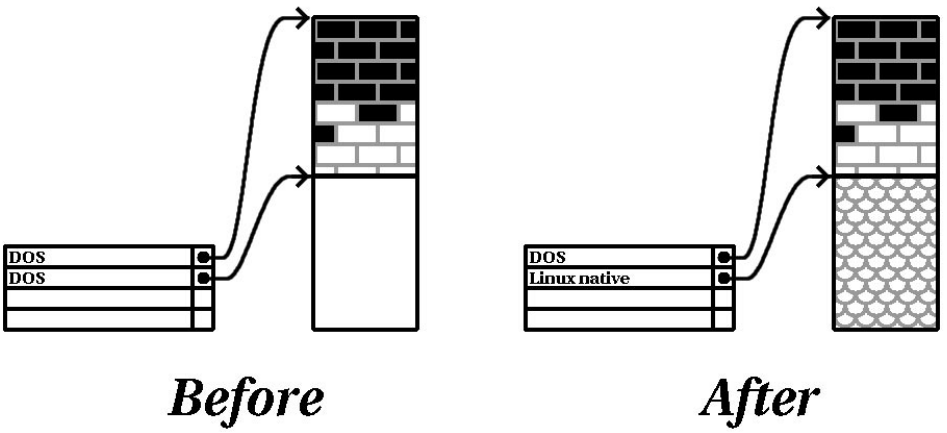


It's important to understand what the resizing software you use does with the newly freed space, so that you can take the appropriate steps. In the case we've illustrated, it would be best to simply delete the new DOS partition, and create the appropriate Linux partition(s).

Create new partition(s)

As the previous step implied, it may or may not be necessary to create new partitions. However, unless your resizing software is Linux-aware, it is likely you will need to delete the partition that was created during the resizing process. Figure B–13, *Disk Drive with Final Partition Configuration* shows this being done.

Figure B–13 Disk Drive with Final Partition Configuration



Note

The following information is specific to Intel-based computers only.

As a convenience to Red Hat Linux users, the DOS `fips` utility is included on the Red Hat Linux/x86 CD 1 in the `dosutils` directory. This is a freely available program that can resize FAT (File Allocation Table) partitions.

WARNING

Many people have successfully used `fips` to resize their hard drive partitions. However, because of the nature of the operations carried out by `fips` and the wide variety of hardware and software configurations under which it must run, Red Hat cannot guarantee that `fips` will work properly on your system. Therefore, no installation support is available for `fips`. *Use it at your own risk.*

That said, if you decide to repartition your hard drive with `fips`, it is *vital* that you do two things:

- *Perform a backup* — Make two copies of all the important data on your computer. These copies should be to removable media (such as tape or diskettes), and you should make sure they are readable before proceeding.
- *Read the documentation* — Completely read the `fips` documentation, located in the `dosutils/fipsdocs` subdirectory on Red Hat Linux/x86 CD 1.

Should you decide to use `fips`, be aware that after `fips` runs you will be left with *two* partitions: the one you resized, and the one `fips` created out of the newly freed space. If your goal is to use that space to install Red Hat Linux, you should delete the newly created partition, either by using `fdisk` under your current operating system or while setting up partitions during a custom-class installation.

B.1.5 Partition Naming Scheme

Linux refers to disk partitions using a combination of letters and numbers which may be confusing, particularly if you're used to the "C drive" way of referring to hard disks and their partitions. In the DOS/Windows world, partitions are named using the following method:

- Each partition's type is checked to determine if it can be read by DOS/Windows.
- If the partition's type is compatible, it is assigned a "drive letter." The drive letters start with a "C" and move on to the following letters, depending on the number of partitions to be labeled.
- The drive letter can then be used to refer to that partition as well as the filesystem contained on that partition.

Red Hat Linux uses a naming scheme that is more flexible and conveys more information than the approach used by other operating systems. The naming scheme is file-based, with filenames in the form:

`/dev/xxyN`

Here's how to decipher the partition naming scheme:

`/dev/`

This string is the name of the directory in which all device files reside. Since partitions reside on hard disks, and hard disks are devices, the files representing all possible partitions reside in `/dev/`.

xx

The first two letters of the partition name indicate the type of device on which the partition resides. You'll normally see either `hd` (for IDE disks) or `sd` (for SCSI disks).

y

This letter indicates which device the partition is on. For example, `/dev/hda` (the first IDE hard disk) or `/dev/sdb` (the second SCSI disk).

N

The final number denotes the partition. The first four (primary or extended) partitions are numbered `1` through `4`. Logical partitions start at `5`. So, for example, `/dev/hda3` is the third primary or extended partition on the first IDE hard disk, and `/dev/sdb6` is the second logical partition on the second SCSI hard disk.

Note

There is no part of this naming convention that is based on partition type; unlike DOS/Windows, *all* partitions can be identified under Red Hat Linux. Of course, this doesn't mean that Red Hat Linux can access data on every type of partition, but in many cases it is possible to access data on a partition dedicated to another operating system.

Keep this information in mind; it will make things easier to understand when you're setting up the partitions Red Hat Linux requires.

B.1.6 Disk Partitions and Other Operating Systems

If your Red Hat Linux partitions will be sharing a hard disk with partitions used by other operating systems, most of the time you'll have no problems. However, there are certain combinations of Linux and other operating systems that require extra care. Information on creating disk partitions compatible with other operating systems is available in several HOWTOs and Mini-HOWTOs, available on the Red Hat Linux Documentation CD in the `HOWTO` and `HOWTO/mini` directories. In particular, the Mini-HOWTOs whose names start with `Linux+` are quite helpful.

Note

If Red Hat Linux/x86 will coexist on your machine with OS/2, you must create your disk partitions with the OS/2 partitioning software —— otherwise, OS/2 may not recognize the disk partitions. During the installation, do not create any new partitions, but do set the proper partition types for your Linux partitions using the Linux `fdisk`.

B.1.7 Disk Partitions and Mount Points

One area that many people new to Linux find confusing is the matter of how partitions are used and accessed by the Linux operating system. In DOS/Windows, it is relatively simple: Each partition gets a "drive letter." You then use the correct drive letter to refer to files and directories on its corresponding partition.

This is entirely different from how Linux deals with partitions and, for that matter, with disk storage in general. The main difference is that each partition is used to form part of the storage necessary to support a single set of files and directories. This is done by associating a partition with a directory through a process known as **mounting**. Mounting a partition makes its storage available starting at the specified directory (known as a **mount point**).

For example, if partition `/dev/hda5` were mounted on `/usr`, that would mean that all files and directories under `/usr` would physically reside on `/dev/hda5`. So the file `/usr/share/doc/FAQ/txt/Linux-FAQ` would be stored on `/dev/hda5`, while the file `/etc/X11/gdm/Sessions/Gnome` would not.

Continuing our example, it is also possible that one or more directories below `/usr` would be mount points for other partitions. For instance, a partition (say, `/dev/hda7`) could be mounted on `/usr/local`, meaning that `/usr/local/man/whatis` would then reside on `/dev/hda7` rather than `/dev/hda5`.

B.1.8 How Many Partitions?

At this point in the process of preparing to install Red Hat Linux, you will need to give some consideration to the number and size of the partitions to be used by your new operating system. The question of "how many partitions" continues to spark debate within the Linux community and, without any end to the debate in sight, it is safe to say that there are probably as many partition layouts as there are people debating the issue.

Keeping this in mind, we recommend that, unless you have a reason for doing otherwise, you should at least create the following partitions:

- *A swap partition* — Swap partitions are used to support virtual memory. In other words, data is written to swap when there is not RAM to hold the data your system is processing. You *must* create a swap partition to correctly use Red Hat Linux. The minimum size of your swap partition should be equal to twice the amount of your computer's RAM or 32 MB, whichever is larger.
- *A `/boot` partition* — The partition mounted on `/boot` contains the operating system kernel (which allows your system to boot Red Hat Linux), along with a few other files used during the bootstrap process.

Make sure you read Section B.1.9, *One Last Wrinkle: Using LILO* — the information there applies to the `/boot` partition!

Due to the limitations of most PC BIOSes, creating a small partition to hold these files is a good idea. This partition should be no larger than 32 MB.

- *A root partition (/)* — The root partition is where `/` (the root directory) resides. In this partitioning layout, all files (except those stored in `/boot`) reside on the root partition. Because of this, it's in your best interest to maximize the size of your root partition. A 1.2 GB root partition will permit the equivalent of a workstation-class installation (with *very* little free space), while a 2.4 GB root partition will let you install every package. Obviously, the more space you can give the root partition, the better.

Specific recommendations concerning the proper size for various Red Hat Linux partitions can be found in the *Official Red Hat Linux x86 Installation Guide*.

B.1.9 One Last Wrinkle: Using LILO

LILO (the LInux LOader) is the most commonly used method to boot Red Hat Linux on Intel-based systems. An operating system loader, LILO operates "outside" of any operating system, using only the Basic I/O System (or BIOS) built into the computer hardware itself. This section describes LILO's interactions with PC BIOSes and is specific to Intel-compatible computers.

BIOS-Related Limitations Impacting LILO

LILO is subject to some limitations imposed by the BIOS in most Intel-based computers. Specifically, most BIOSes can't access more than two hard drives, and they can't access any data stored beyond cylinder 1023 of any drive. Note that some recent BIOSes do not have these limitations, but this is by no means universal.

All the data LILO needs to access at boot time (including the Linux kernel) are located in the `/boot` directory. If you follow the partition layout recommended above or are performing a workstation- or server-class install, the `/boot` directory will be in a small, separate partition. Otherwise, it may reside in the root partition (`/`). In either case, the partition in which `/boot` resides must conform to the following guidelines if you are going to use LILO to boot your Red Hat Linux system:

On First Two IDE Drives

If you have 2 IDE (or EIDE) drives, `/boot` must be located on one of them. Note that this two-drive limit also includes any IDE CD-ROM drives on your primary IDE controller. So, if you have one IDE hard drive, and one IDE CD-ROM on your primary controller, `/boot` must

be located on the first hard drive *only*, even if you have other hard drives on your secondary IDE controller.

On First IDE or First SCSI Drive

If you have one IDE (or EIDE) drive and one or more SCSI drives, `/boot` must be located either on the IDE drive or the SCSI drive at ID 0. No other SCSI IDs will work.

On First Two SCSI Drives

If you have only SCSI hard drives, `/boot` must be located on a drive at ID 0 or ID 1. No other SCSI IDs will work.

Partition *Completely* Below Cylinder 1023

No matter which of the above configurations apply, the partition that holds `/boot` must be located entirely below cylinder 1023. If the partition holding `/boot` straddles cylinder 1023, you may face a situation where LILO will work initially (because all the necessary information is below cylinder 1023) but will fail if a new kernel is to be loaded and that kernel resides above cylinder 1023.

As mentioned earlier, it is possible that some of the newer BIOSes may permit LILO to work with configurations that don't meet these guidelines. Likewise, some of LILO's more esoteric features may be used to get a Linux system started, even if the configuration doesn't meet our guidelines. However, due to the number of variables involved, Red Hat cannot support such efforts.

Note

Disk Druid, as well as the workstation- and server-class installations, takes these BIOS-related limitations into account.

C Driver Disks

C.1 Why Do I Need a Driver Disk?

While the Red Hat Linux installation program is loading, you may see a screen that asks you for a driver disk. The driver disk screen is most often seen in three scenarios:

- If you are running the installation program in `expert mode`
- If you run the installation program by entering **`linux dd`** at the `boot:` prompt
- If you run the installation program on a computer which does not have any PCI devices

C.1.1 So What Is a Driver Disk Anyway?

A driver disk adds support for hardware that is not otherwise supported by the installation program. The driver disk could be produced by Red Hat, it could be a disk you make yourself from drivers found on the Internet, or it could be a disk that a hardware vendor includes with a piece of hardware.

There is really no need to use a driver disk unless you need a particular device in order to install Red Hat Linux. You will most likely use a driver disk for non-standard or very new CD-ROM drives, SCSI adapters or NICs. These are the only devices used during the installation that might require drivers not included on the Red Hat Linux CD-ROMs (or floppy disk, if you created an installation boot floppy to begin the install process).

Note

If an unsupported device is not needed to install Red Hat Linux on your system, continue with a regular installation and then add support for the new piece of hardware once the installation is complete.

C.1.2 How Do I Obtain a Driver Disk?

The Red Hat Linux CD-ROM 1 includes a driver disk image (`images/drivers.img`) containing many rarely used drivers. If you suspect that your system may require one of these drivers, it may be a good idea to go ahead and create the driver disk floppy before beginning your Red Hat Linux installation.

Another option for finding specialized driver disk information is on Red Hat's website at http://www.redhat.com/support/errata under the section called **Bug Fixes**. Occasionally, very popular hardware may be made available after a release of Red Hat Linux that will not work with drivers already in the installation program or included on the driver disk image on the Red Hat Linux

CD-ROM 1. In cases like this, the Red Hat website may contain a link to a driver disk image you can use to install Red Hat Linux using that hardware.

Creating a Driver Disk from an Image File

If you have a driver disk image that you need to write to a floppy disk, this can be done from within DOS or Red Hat Linux.

To create a driver disk from a driver disk image using Red Hat Linux:

1. Insert a blank, formatted floppy disk into the first floppy drive.
2. From the same directory containing the driver disk image, such as *dd.img*, type `cat dd.img > /dev/fd0` as root.

To create a driver disk from a driver disk image using DOS:

1. Insert a blank, formatted floppy disk into the a: drive.
2. From the same directory containing the driver disk image, such as *dd.img*, type `rawrite dd.img a:` at the command line.

C.1.3 Using a Driver Disk During Installation

Simply having a driver disk is not enough. You must specifically tell the Red Hat Linux installation program to load that driver disk and use it during the installation process.

Note

A driver disk is different than a boot disk. If you require a boot floppy to begin the Red Hat Linux installation on your system, you will still need to create that floppy and boot from it before using your driver disk.

If you do not already have a installation floppy disk and your system does not support booting from the CD-ROM, create an installation floppy disk using the correct *filename*`.img` file (such as `boot.img`) on the Red Hat Linux CD-ROM 1 in the `images` directory. For instructions on how to make a boot disk, see the *Official Red Hat Linux x86 Installation Guide* section called *Making Installation Diskettes*.

Once you have created your driver disk, begin the installation process by booting from the Red Hat Linux CD-ROM 1 (or the installation boot floppy you made if you cannot boot from the CD-ROM for whatever reason). Then, at the `boot:` prompt, enter either **`linux expert`** or **`linux dd`**.

The Red Hat Linux installation program will ask you to insert the driver disk. Once the driver disk is read by the installer, it can then apply those drivers to hardware discovered on your system later in the installation process.

D RAID (Redundant Array of Independent Disks)

D.1 What is RAID?

The basic idea behind RAID is to combine multiple small, inexpensive disk drives into an array to accomplish performance or redundancy goals not attainable with one large and expensive drive. This array of drives will appear to the computer as a single logical storage unit or drive.

RAID is a method in which information is spread across several disks, using techniques such as **disk striping** (RAID Level 0), **disk mirroring** (RAID level 1), and **disk striping with parity** (RAID Level 5) to achieve redundancy, lower latency and/or increase bandwidth for reading or writing to disks, and maximize the ability to recover from hard disk crashes.

The underlying concept of RAID is that data may be distributed across each drive in the array in a consistent manner. To do this, the data must first be broken into consistently-sized "chunks" (often 32K or 64K in size, although different sizes can be used). Each chunk is then written to a hard drive in RAID according to the RAID level used. When the data is to be read, the process is reversed, giving the illusion that multiple drives are actually one large drive.

D.1.1 Who Should Use RAID?

Anyone who needs to keep large quantities of data on hand (such as an average system administrator) would benefit by using RAID technology. Primary reasons to use RAID include:

- Enhanced speed
- Increased storage capacity using a single virtual disk
- Lessening the impact of a disk failure

D.1.2 RAID: Hardware vs. Software

There are two possible RAID approaches: Hardware RAID and Software RAID.

Hardware RAID

The hardware-based system manages the RAID subsystem independently from the host and presents to the host only a single disk per RAID array.

An example of a Hardware RAID device would be one that connects to a SCSI controller and presents the RAID arrays as a single SCSI drive. An external RAID system moves all RAID handling "intelligence" into a controller located in the external disk subsystem. The whole subsystem is connected to the host via a normal SCSI controller and appears to the host as a single disk.

RAID controllers also come in the form of cards that *act* like a SCSI controller to the operating system but handle all of the actual drive communications themselves. In these cases, you plug the drives into the RAID controller just like you would a SCSI controller, but then you add them to the RAID controller's configuration, and the operating system never knows the difference.

Software RAID

Software RAID implements the various RAID levels in the kernel disk (block device) code. It offers the cheapest possible solution, as expensive disk controller cards or hot-swap chassis [1] are not required. Software RAID also works with cheaper IDE disks as well as SCSI disks. With today's fast CPUs, Software RAID performance can excel against Hardware RAID.

The MD driver in the Linux kernel is an example of a RAID solution that is completely hardware independent. The performance of a software-based array is dependent on the server CPU performance and load.

For information on configuring Software RAID in the Red Hat Linux installation program, refer to the *Official Red Hat Linux Customization Guide*.

For those interested in learning more about what Software RAID has to offer, here is a brief list of the most important features:

- Threaded rebuild process
- Fully kernel-based configuration
- Portability of arrays between Linux machines without reconstruction
- Backgrounded array reconstruction using idle system resources
- Hot-swappable drive support
- Automatic CPU detection to take advantage of certain CPU optimizations

D.1.3 RAID Levels and Linear Support

RAID supports various configurations, including levels 0, 1, 4, 5, and linear. These RAID types are defined as follows:

- *Level 0* — RAID level 0, often called "striping," is a performance-oriented striped data mapping technique. This means the data being written to the array is broken down into strips and written across the member disks of the array, allowing high I/O performance at low inherent cost but provides no redundancy. The storage capacity of a level 0 array is equal to the total capacity of the member disks in a Hardware RAID or the total capacity of member partitions in a Software RAID.

[1] A hot-swap chassis allows you to remove a hard drive without having to power-down your system.

- *Level 1* — RAID level 1, or "mirroring," has been used longer than any other form of RAID. Level 1 provides redundancy by writing identical data to each member disk of the array, leaving a "mirrored" copy on each disk. Mirroring remains popular due to its simplicity and high level of data availability. Level 1 operates with two or more disks that may use parallel access for high data-transfer rates when reading but more commonly operate independently to provide high I/O transaction rates. Level 1 provides very good data reliability and improves performance for read-intensive applications but at a relatively high cost[2]. The storage capacity of the level 1 array is equal to the capacity of one of the mirrored hard disks in a Hardware RAID or one of the mirrored partitions in a Software RAID.

- *Level 4* — Level 4 uses parity[3]concentrated on a single disk drive to protect data. It's better suited to transaction I/O rather than large file transfers. Because the dedicated parity disk represents an inherent bottleneck, level 4 is seldom used without accompanying technologies such as write-back caching. Although RAID level 4 is an option in some RAID partitioning schemes, it is not an option allowed in Red Hat Linux RAID installations[4]. The storage capacity of Hardware RAID level 4 is equal to the capacity of member disks, minus the capacity of one member disk. The storage capacity of Software RAID level 4 is equal to the capacity of the member partitions, minus the size of one of the partitions if they are of equal size.

- *Level 5* — This is the most common type of RAID. By distributing parity across some or all of an array's member disk drives, RAID level 5 eliminates the write bottleneck inherent in level 4. The only performance bottleneck is the parity calculation process. With modern CPUs and Software RAID, that usually isn't a very big problem. As with level 4, the result is asymmetrical performance, with reads substantially outperforming writes. Level 5 is often used with write-back caching to reduce the asymmetry. The storage capacity of Hardware RAID level 5 is equal to the capacity of member disks, minus the capacity of one member disk. The storage capacity of Software RAID level 5 is equal to the capacity of the member partitions, minus the size of one of the partitions if they are of equal size.

- *Linear RAID* — Linear RAID is a simple grouping of drives to create a larger virtual drive. In linear RAID, the chunks are allocated sequentially from one member drive, going to the next drive

[2] RAID level 1 comes at a high cost because you write the same information to all of the disks in the array, which wastes drive space. For example, if you have RAID level 1 set up so that your root (/) partition exists on two 40G drives, you have 80G total but are only able to access 40G of that 80G. The other 40G acts like a mirror of the first 40G.

[3] Parity information is calculated based on the contents of the rest of the member disks in the array. This information can then be used to reconstruct data when one disk in the array fails. The reconstructed data can then be used to satisfy I/O requests to the failed disk before it is replaced and to repopulate the failed disk after it has been replaced.

[4] RAID level 4 takes up the same amount of space as RAID level 5, but level 5 has more advantages than level 4. For this reason, level 4 is not supported.

only when the first is completely filled. This grouping provides no performance benefit, as it is unlikely that any I/O operations will be split between member drives. Linear RAID also offers no redundancy and, in fact, decreases reliability —— if any one member drive fails, the entire array cannot be used. The capacity is the total of all member disks.

E PowerTools

E.1 What are PowerTools?

Red Hat PowerTools is a collection of software packages built for the Red Hat Linux 7.1 operating system. PowerTools includes the latest versions (as of the Red Hat Linux release date) of hundreds of programs — so finding an interesting application should be easy.

Among the many applications are audio programs, chat clients, development tools, editors, file managers, emulators, games, graphics programs, productivity applications, math/statistics packages, systems administration utilities, network management tools, and window managers.

Are you a system administrator? PowerTools features an array of tools that can make your life easier and possibly replace several expensive diagnostic utilities with a common application. Take a look at applications such as Ethereal for analyzing network protocols, PortSentry to stop port scanners on your network, or Postfix as an alternative to Sendmail.

Love playing games? PowerTools contains a number of very fun, basic games, such as SpeedX, XFrisk, and Amphetamine.

And since installing and uninstalling software packages on Red Hat Linux is easy using RPM or Gnome-RPM, you can quickly try out different applications that do the same thing before deciding upon the one that is best for you.

E.2 PowerTools Packages

If you already know of a PowerTools package that you would like to install, see Section E.3, *Installing PowerTools Packages* for installation information.

However, due to the large number of PowerTools packages available, it is helpful to be able to search through the package descriptions to find those that meet your requirements.

E.2.1 Reading the Contents of the CD-ROM

You can read the contents of the PowerTools CD-ROM from a shell prompt (either in a terminal window or in console mode). First you have to mount the CD-ROM drive.

Mounting the PowerTools CD-ROM

If your system is not set up to automount the CD-ROM drive when a CD is inserted, place the PowerTools CD in your CD-ROM drive. As root, type the following:

```
mount -t iso9660 /dev/cdrom /mnt/cdrom
```

Note

On your system, you or the system administrator may already allow users (instead of only root) to mount the CD-ROM drive. Users have this privilege if the `user` option is included in the `/dev/cdrom` line in the `/etc/fstab` file. However, keep in mind that you must be logged in as root to install any PowerTools RPMs.

Navigating the `CONTENTS` File

After you have mounted the drive, `cd` to the mounted CD-ROM directory with the following command:

```
cd /mnt/cdrom
```

Finally, type `less CONTENTS` to view the available applications. The `CONTENTS` file contains every program on the PowerTools CD-ROM, listed in alphabetical order.

Reading the `CONTENTS` file on the PowerTools CD-ROM can be a daunting task, considering the sheer number of applications available. Here are a few tricks to find a particular type of program without having to read through all of the descriptions:

- *Use the Group name* — Every application is assigned to a particular group. For example, FaxMail, a fax sending utility, is in the Applications/Communications group, and Icecast, an MP3 Internet broadcasting system, is in the Applications/Multimedia group. By skimming the group names, you can save the time of having to read each package's description.
- *Search using keywords* — The `ls` command supports easy searching. If you know you are looking for an IRC client, you can type `less CONTENTS` to view `CONTENTS` and then type **`/IRC`** and press [Enter]. You will be taken to the first IRC client in the list. If this one does not interest you, pressing the [n] key repeatedly will let you skim through the `CONTENTS` file, looking only at IRC-related packages.

If you have trouble using the `less` command, type `man less` at a prompt for help.

Unmounting the PowerTools CD-ROM

When you are finished using the PowerTools CD-ROM to install packages, you can remove it from your CD-ROM drive. If you have the CD-ROM mounted in the `/mnt/cdrom` directory, do the following:

1. Change directories using `cd /mnt` so that you are one level above the `/mnt/cdrom` directory.
2. Type `umount /mnt/cdrom` to unmount the CD-ROM.

3. Type `eject /dev/cdrom` and the CD-ROM drive will open so that you can remove the CD.

E.3 Installing PowerTools Packages

E.3.1 Installing PowerTools in a GUI Environment

If you're using GNOME or KDE, place the CD-ROM in your CD-ROM drive. You'll be prompted for the root password (you must be root in order to install packages). After you type in the root password, either the Gnome-RPM or the Kpackage package management program will start automatically (depending on you GUI environment) and can be used to install PowerTools.

Refer to the *Official Red Hat Linux Getting Started Guide* for specific instructions on how to use Gnome-RPM. See http://www.general.uwa.edu.au/u/toivo/kpackage for more information on how to use Kpackage.

If you are not using GNOME or KDE, you will need to use the shell prompt to install PowerTools.

E.3.2 Installing PowerTools from the Shell Prompt

First, mount the PowerTools CD-ROM on your CD-ROM drive and use `ls` to view its contents. If you need to know how to mount a CD-ROM, see *Mounting the PowerTools CD-ROM* in Section E.2.1.

You will see the following directories: `SRPMS` and `RedHat`. The `SRPMS` directory contains the PowerTools source RPMs. The `RedHat/RPMS` directory contains the RPMs for the three specified operating system architectures.

The `RedHat/RPMS` path is used as a general example. You should substitute the correct directory for `RedHat/RPMS`, depending upon your architecture and which package you're installing.

`cd` to the `RedHat/RPMS` directory:

```
cd RedHat/RPMS
```

List the files in the directory with `ls` to see the complete list of RPM packages included for Intel-compatible systems.

You will probably want more information about a specific package before you can decide whether you want to install it. You can use RPM's querying capability to find out more information about the packages, such as the packages' functions and origination. See *Official Red Hat Linux Customization Guide* for detailed instructions on how to query packages using RPM.

Alternatively, you can search through the `CONTENTS` file to find packages that interest you. See *Navigating the `CONTENTS` File* in Section E.2.1 for instructions on how to do this.

You can install your selected packages with RPM. RPM is a powerful command line-driven package management system. See *Official Red Hat Linux Customization Guide* for more information on how to use RPM to install and manage PowerTools packages.

Once you have finished installing your packages, you should unmount your CD-ROM. If you do not already know how to unmount the CD-ROM drive, see *Unmounting the PowerTools CD-ROM* in Section E.2.1.

E.4 Uninstalling PowerTools

To uninstall PowerTools packages from your system, you simply remove them in the same way any other RPM-installed package is removed.

First, you have to know the name of the package you would like to uninstall. For example, if you know you want to remove `thrust-0.83c-11` from your system, type as root:

```
rpm -e thrust
```

In general, `rpm -e <packagename>` will remove the package and its related files from your system. The PowerTools CD-ROM is not required for this operation.

For more information concerning the use of RPM, see *Official Red Hat Linux Customization Guide*.

Index

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

R

S

T

U

V

W