Securing Intel-Based Red Hat Linux 7.3 Computers

Peter Andrew Schwenk
GSEC 1.4 Option 1

Introduction

Red Hat Linux is by far the most popular of the commercial GNU/Linux distributions available today. It is known for its ease of installation and maintenance. Though it had a reputation for being a bit insecure out-of-the-box, Red Hat has worked at making the latest versions much better at initial security. Version 7.3, the latest non-beta version of the Red Hat Linux as of this writing, has an installation program that goes a long way toward making it secure, however, there is still quite a bit to be done by the administrator to enhance security.

This article will focus solely on Version 7.3 running on Intel-based computers, but much of what is presented here is applicable to earlier versions in the 7.x line and non-Intel-based architectures. Provided in this article are some basic guidelines and instructions on making a Red Hat Linux 7.3 workstation or server as secure as possible, given the intended use for the computer. Red Hat Linux 7.3 makes available many tools that are useful for keeping a system in an extremely secure state, and this article will discuss how to use them effectively. Methods for securely configuring the various services that a Red Hat Linux 7.3 computer may offer, such as web serving with Apache[1] or file serving with Samba[2], are broad topics by themselves and will not be covered in this article.

Preliminary Issues

Prior to putting any computer into service, it is important to evaluate the environment into which it will go. Here are some important questions to answer:

- What does the organization's security policy say about what parts of an operating system can be installed in this particular case?
- Who will be using the system and what type of access will they need?
- Where will the system reside in your network?
- For what will the computer be used?
- Who will be administering the system?
- Who will be responsible for the computer's security?

The organization's security policy should have these clearly answered.

Physical Security and the BIOS

Any software security measures that one may take are immaterial if an attacker that has physical access to the computer can compromise it. The bulk of

[1] http://www.apache.org
[2] http://www.samba.org

physical security is beyond the scope of this article, so this section will focus on those aspects of it that can be addressed with settings in the computer's BIOS. This is not to say that other aspects of physical security are unimportant. In fact, it is necessary to consider that an intruder removing a battery or connecting the correct pins on the main system circuit board can frequently wipe the BIOS settings, including passwords, from the computer's memory[3].

While not strictly a Red Hat Linux concern, the BIOS configuration usually contains settings that can be employed to protect the installed operating system(s). The primary settings in the BIOS that are available for protecting the installed operating systems from harm are:

- Boot and BIOS maintenance passwords
- Boot device restrictions

BIOS Passwords

Most modern BIOSes have the ability to assign a password for both booting the computer and for BIOS administration. When a boot password is assigned, it must be supplied before the computer will boot. The BIOS administration password must be entered before the computer will allow anyone into the BIOS setup utility.

Obviously, whether or not one would assign either of these passwords depends heavily on the particular situation. Boot passwords can be a big annoyance for both the administrator and the user. The administrator would need to have a procedure in place for assigning and entering the passwords into each system under his or her control. A user would have one more password to remember. BIOS administration passwords are almost always a good idea because not assigning one is, in effect, giving away the keys to the castle because it is the only gatekeeper from the rest of the BIOS settings, security-related or otherwise.

Boot Device Restrictions

In many situations, it can be beneficial to keep a user of a computer from being able to boot from a device other than the hard drive, like a floppy disk or CDROM. For example, if someone were to boot a Red Hat Linux computer with the installation CDROM into Rescue Mode, they would effectively wield administrative power over the system. Here is a list of examples of what kind of damage one could inflict with a Red Hat Linux installation CDROM:

- Change the *root* (the user that has absolute administrative control over the system) user password to something they know or remove the password altogether
- Remove or modify any file stored on disk
- Re-install the operating system, essentially destroying all pre-existing files on the disks

[3] http://www.zephirtech.com/misc/biospasswords.html

In light of these examples, it is easy to see why it is a good idea to restrict booting to the hard disk.

In concert with BIOS passwords, boot device restrictions significantly enhance the physical security of a system against most casual intruders.

Installation

The first task involved with securing any Red Hat Linux is getting the software installed. There are a few security considerations to be taken into account when installing the software. They are:

- Install in a safe environment
- What to install
- GRUB boot loader password
- Firewall settings
- Authentication services
- *root* password

This article will discuss the security-related aspects of the installation process and will not walk the reader, step-by-step through the installation process. For detailed installation instructions, please consult the Red Hat Linux 7.3 Installation Guide[4].

Install in a Safe Environment

Even if the system onto which Red Hat Linux 7.3 is being installed is on a protected network (e.g. on a private network or behind a firewall), it is good practice to perform the installation with the system disconnected from the network. It is possible that an intruder could happen upon the preconfigured system and compromise it before the administrator has a chance to secure it. It is up to the organization's security policy to dictate whether this practice is followed.

What to Install

It is important to consider what to install before doing it because one wants to give a potential intruder as few avenues for exploiting the system as possible. For this reason, it is a good idea to install the least software necessary to provide the services that the system is intended to provide. Of course, this takes quite a bit of forethought as Red Hat provides a plethora of software to install. It also depends on the policies and procedures in place at the organization into which the computer will be installed.

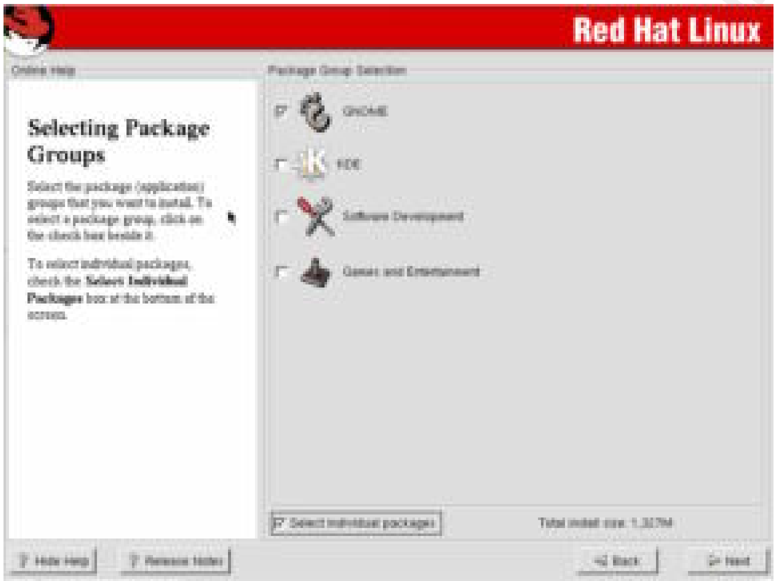
Red Hat provides five system type installation classes from which one can choose to ease the installation process. These are:

1. Workstation
2. Server

[4] http://www.redhat.com/docs/manuals/linux/RHL-7.3-Manual/install-guide/

3. Laptop
4. Custom
5. Upgrade

Everything is a special case of the **Custom** installation class. This obviously installs every piece of software that comes with the distribution. No matter which installation class is chosen, the administrator has the opportunity to fine-tune the installation by choosing the "select individual packages" option as illustrated in Figure 1.



Selecting Package Groups

Select the package (application) groups that you want to install. To select a package group, click on the check box beside it.

To select individual packages, check the **Select Individual Packages** box at the bottom of the screen.

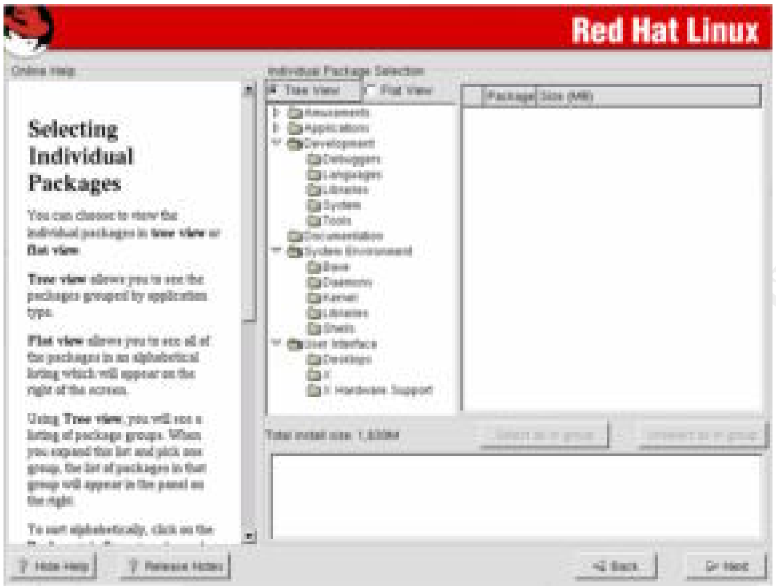
☑ GNOME
☐ KDE
☐ Software Development
☐ Games and Entertainment

☑ Select individual packages

Total install size: 1,327M

Figure 1 - Package Group Selection Screen with Select Individual Packages Setting

This allows for very fine-grained control over which packages get installed. There are two ways to view the list of packages from which one can choose, *tree view* (see Figure 2) and *flat view* (see Figure 3).



Selecting Individual Packages

You can choose to view the individual packages in **tree view** or **flat view**.

Tree view allows you to see the packages grouped by application type.

Flat view allows you to see all of the packages in an alphabetical listing which will appear on the right of the screen.

Using **Tree view**, you will see a listing of package groups. When you expand this list and pick one group, the list of packages in that group will appear in the panel on the right.

To sort alphabetically, click on the

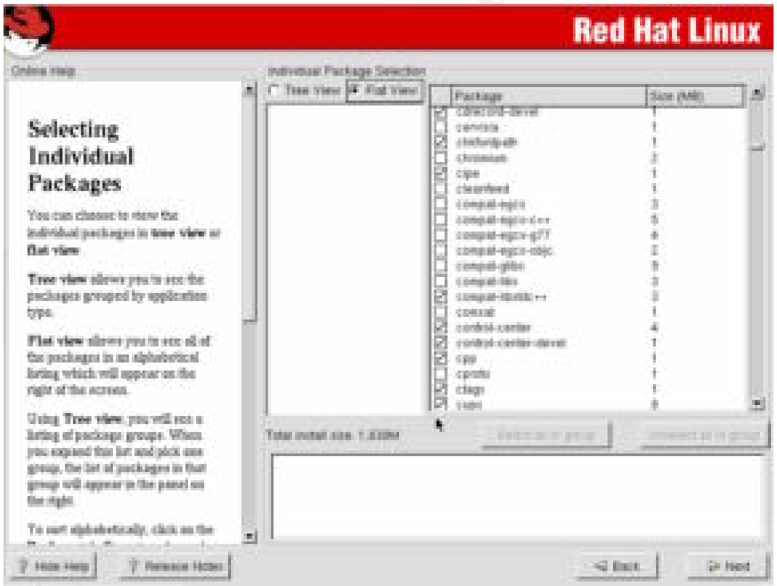
● Tree View ○ Flat View

- Amusements
- Applications
- Development
  - Debuggers
  - Languages
  - Libraries
  - System
  - Tools
- Documentation
- System Environment
  - Base
  - Daemons
  - Kernel
  - Libraries
  - Shells
- User Interface
  - Desktops
  - X
  - X Hardware Support

	Package	Size (MB)

Total install size: 1,830M

Figure 2 - Viewing Packages to Install with Tree View



Selecting Individual Packages

You can choose to view the individual packages in **tree view** or **flat view**.

Tree view allows you to see the packages grouped by application type.

Flat view allows you to see all of the packages in an alphabetical listing which will appear on the right of the screen.

Using **Tree view**, you will see a listing of package groups. When you expand this list and pick one group, the list of packages in that group will appear in the panel on the right.

To sort alphabetically, click on the

○ Tree View ● Flat View

	Package	Size (MB)
☑	cdrecord-devel	1
□	cervisia	1
☑	chkfontpath	1
□	chromium	2
☑	cipe	1
□	cleanfeed	1
□	compat-egcs	3
□	compat-egcs-c++	5
□	compat-egcs-g77	6
□	compat-egcs-objc	2
□	compat-glibc	5
□	compat-libs	2
☑	compat-libstdc++	2
□	[illegible]	1
☑	control-center	4
☑	control-center-devel	1
☑	cpp	1
□	cproto	1
☑	ctags	1
☑	[illegible]	8

Total install size: 1,830M

Figure 3 - Viewing Packages to Install with Flat View

Tree view organizes the list into categories to ease the chore of locating packages via their purposes. List view provides an alphabetical list of all packages for those who know exactly what they want by name.

GRUB Boot Loader Password

A boot loader is a small program usually stored on disk that is charged with the duty of booting the rest of the operating system. An example of a very simple boot loader is DOS's Master Boot Record (MBR) that gets written to a bootable floppy or hard disk with MS-DOS-based operating systems installed on them.

Traditionally, the boot loader used with many distributions of Linux is called LILO, which is a contraction of Linux Loader. It is a relatively full-featured boot loader that provides a configurable boot menu for running multiple operating systems and a way to adjust kernel parameters. One problem with LILO is that it is not possible to keep a user with direct access to the console from booting the Linux kernel any way they want, including in single user (administrative) mode, by hand-crafting a boot command at the LILO prompt. A user that is able to get the system into single user mode is effectively *root* and wields the same power as described in the **Boot Device Restrictions** section above.

As of Version 7.2, Red Hat Linux supplies an improved alternative boot loader, called GRUB, which solves the problem described above. GRUB optionally password-protects the selections in the boot menu. It also does not provide a prompt into which users can type free-form boot commands. While this may seem like an inconvenience in some circumstances, it goes a long way for system security.

Firewall Configuration During Installation

Red Hat Linux 7.3 comes with the 2.4.18 version kernel. The 2.4.x series kernels have a new packet-filtering sub-system called Netfilter (also known as *iptables*)[5] in addition to the previous incarnation, *ipchains*[6]. The advantages that the Netfilter packet filtering technology has over ipchains are[7]:

- Stateful packet filtering
- Full-featured Network Address Translation (NAT)
- More flexible filtering on more packet fields
- Additional features as modules

Stateful packet filtering means that connection tracking can be done instead of just filtering on individual packet properties. For example, Netfilter can watch connection handshakes and drop the connection after it meets criteria specified by the administrator. *Stateless* packet filters, like ipchains, have no memory of prior packets in a connection handshake, so they can only act on properties of individual packets.

Red Hat provides basic control over packet filtering during installation using its Lokkit configuration tool (see Figure 4), which actually configures ipchains rules. Lokkit provides 3 levels of firewall protection[8]:

[5] http://netfilter.samba.org

[6] http://netfilter.samba.org/ipchains/ and http://www.tldp.org/HOWTO/IPCHAINS-HOWTO.html

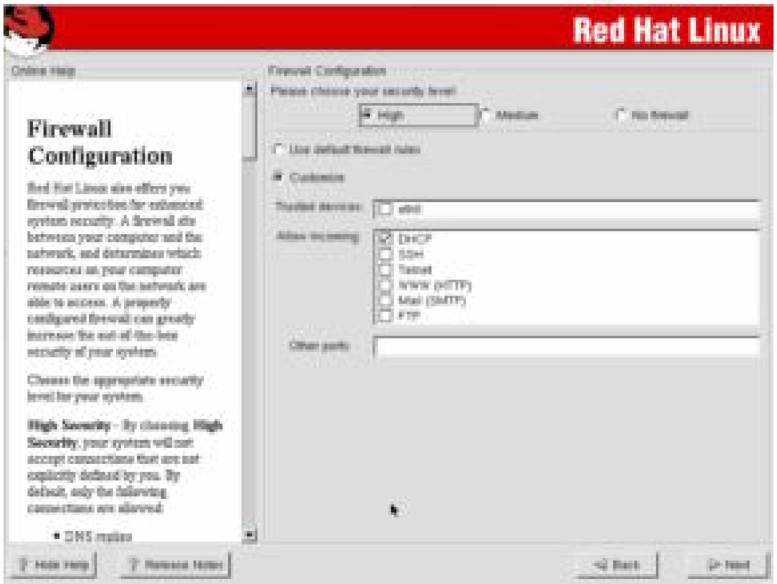
[7] http://www.oofle.com/iptables/comparison.htm

[8] http://www.redhat.com/docs/manuals/linux/RHL-7.3-Manual/install-guide/s1-firewallconfig.html

None All packets are allowed. This provides absolutely no protection and should not be used.

Medium All incoming traffic to privileged (less than 1024) ports is disallowed with the exception of DNS replies and DHCP-related traffic. All non-privileged port traffic is allowed except for NFS (port 2049), X Windows (port 6000), and the X Font Server (xfs – port 7100). The administrator using the Customize option in the Firewall Configuration screen must explicitly configure any other exceptions.

High Disallows all incoming connections not explicitly allowed. The default exceptions are DNS replies and DHCP-related packets. The administrator must choose exactly what IP ports to let through the firewall. The Customize option in the Firewall Configuration screen is used to define any exceptions.



Firewall Configuration

Red Hat Linux also offers you firewall protection for enhanced system security. A firewall sits between your computer and the network, and determines which resources on your computer remote users on the network are able to access. A properly configured firewall can greatly increase the out-of-the-box security of your system.

Choose the appropriate security level for your system.

High Security - By choosing **High Security**, your system will not accept connections that are not explicitly defined by you. By default, only the following connections are allowed:

- DNS replies

Firewall Configuration

Please choose your security level: ● High ○ Medium ○ No firewall

○ Use default firewall rules

● Customize

Trusted devices: □ eth0

Allow incoming:
☑ DHCP
□ SSH
□ Telnet
□ WWW (HTTP)
□ Mail (SMTP)
□ FTP

Other ports:

Figure 4 - Firewall Configuration During Installation

For most situations, the firewall configuration at installation will be sufficient. For those situations where it is not, Red Hat Linux 7.3 provides tools for setting up either more elaborate ipchains rules or Netfilter rules. Complicated firewall configuration is a subject that could fill a volume by itself, so it will not be discussed here. Readers who are interested in the tools that Red Hat provides for configuring complicated firewalls should read http://www.redhat.com/docs/manuals/linux/RHL-7.3-Manual/ref-guide/ch-iptables.html.

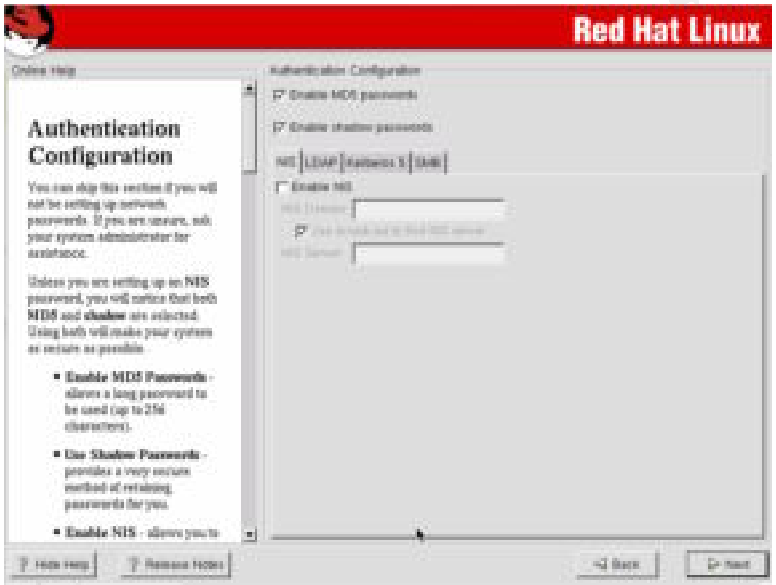
Authentication Configuration

There are many user authentication settings to adjust during the installation of Red Hat Linux 7.3. The options are very comprehensive and should accommodate almost every common authentication need:

- Enable MD5 Passwords

- Enable Shadow Passwords
- NIS Authentication
- LDAP Authentication
- Kerberos 5 Authentication
- SMB Authentication

The Authentication Configuration screen (see Figure 5) only appears if the Custom installation class is chosen. If Workstation, Server, or Laptop installs are performed, only MD5 passwords and shadow passwords are enabled by default, and the others can be turned on as needed after the system is running. A summary of each setting will be provided in this section.



Authentication Configuration

You can skip this section if you will not be setting up network passwords. If you are unsure, ask your system administrator for assistance.

Unless you are setting up an NIS password, you will notice that both MD5 and shadow are selected. Using both will make your system as secure as possible.

- **Enable MD5 Passwords** - allows a long password to be used (up to 256 characters).
- **Use Shadow Passwords** - provides a very secure method of retaining passwords for you.
- **Enable NIS** - allows you to

Authentication Configuration

☑ Enable MD5 passwords

☑ Enable shadow passwords

☐ Enable NIS

NIS Domain:

☑ Use broadcast to find NIS server

NIS Server:

Figure 5 - Authorization Configuration Screen

Enable MD5 Passwords

There are two hash functions that are normally used to encrypt user passwords on UNIX-like operating systems like Red Hat Linux 7.3: DES and MD5. DES is weaker than MD5, but it might be necessary to use it for compatibility with other operating systems, like Solaris, Sun Microsystem's UNIX OS. DES also restricts password length to 8 characters, whereas MD5 allows for 256. If there are no compatibility requirements with which to contend, MD5 is the better choice.

Enable Shadow Passwords

The user name and password database is normally a text file called **/etc/passwd**. This file needs to be readable by all users of the system for software to have the ability to look up user names and IDs. It used to be that this file would also contain the encrypted passwords, which produced a security problem because it provided the opportunity for malicious users to try to crack them. Readers who are unfamiliar with the file permission scheme that Linux and UNIX use should read http://www.redhat.com/docs/manuals/linux/RHL-7.3-

Manual/getting-started-guide/s1-navigating-ownership.html for a brief explanation.

This problem was solved by creating a second companion file called **/etc/shadow** to store the encrypted passwords. This file is set to be readable only by the *root* user who owns the file. The **/etc/passwd** file is modified to only contain placeholders where the encrypted password would normally be. Separating out the sensitive information into another file with limited access lessens the possibilities of a malicious user getting access to the encrypted passwords.

NIS Authentication

NIS, also known as YP (for Yellow Pages), is a simple server-based database system that is mainly used to allow many computers access to the same set of authentication data, like user names and passwords. NIS servers answer queries from computers in the *domain* that they serve, so NIS clients need to be set up with the same domain name. NIS clients can either be configured to use a specific NIS server using its IP address or to broadcast their requests for any available server to receive. It is desirable to specify the server to preclude the possibility of a rogue NIS server supplying phony information as a step in some sort of exploit. Figure 5 illustrates the NIS tab of the Authentication Configuration screen into which the administrator can specify the configuration details of an NIS client.

NIS is not a very secure protocol. There is minimal client or server authentication other than trusting IP addresses. By knowing the domain, it is trivial to retrieve the password table (*map* in NIS-speak) with the encrypted passwords.

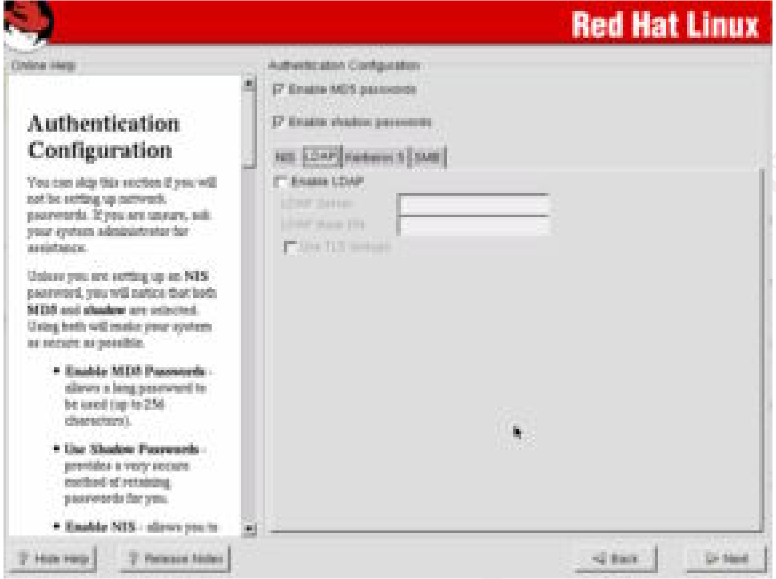
If the Red Hat Firewall Configuration is set to either Medium or High, NIS will not work because the necessary ports are filtered out[9].

LDAP[10] Authentication

LDAP (Light-weight Directory Access Protocol) is another way to provide user authentication and other information to many clients. To access the data in the server, the client must be configured to query a particular server via its IP address. The database is structured in a tree-like, hierarchical fashion[11], so the client must specify the path from the root to the node where queries will start. This starting-point node is called the *Base Distinguishing Name* (DN). The transport layer communication between the client and server can be encrypted using the *Transport Layer Security* (TLS) protocol. Figure 6 shows the entry fields of the LDAP Tab of the Authentication Configuration screen.

[9] http://www.redhat.com/docs/manuals/linux/RHL-7.3-Manual/install-guide/s1-firewallconfig.html
[10] http://www.openldap.org
[11] http://www.openldap.org/doc/admin21/intro.html

If the Red Hat Firewall Configuration is set to either Medium or High, LDAP will not work because the necessary ports are filtered out.



Authentication Configuration

You can skip this section if you will not be setting up network passwords. If you are unsure, ask your system administrator for assistance.

Unless you are setting up an **NIS** password, you will notice that both **MD5** and **shadow** are selected. Using both will make your system as secure as possible.

- **Enable MD5 Passwords** - allows a long password to be used (up to 256 characters).
- **Use Shadow Passwords** - provides a very secure method of retaining passwords for you.
- **Enable NIS** - allows you to

☑ Enable MD5 passwords

☑ Enable shadow passwords

NIS | LDAP | Kerberos 5 | SMB

□ Enable LDAP

LDAP Server:

LDAP Base DN:

□ Use TLS lookups

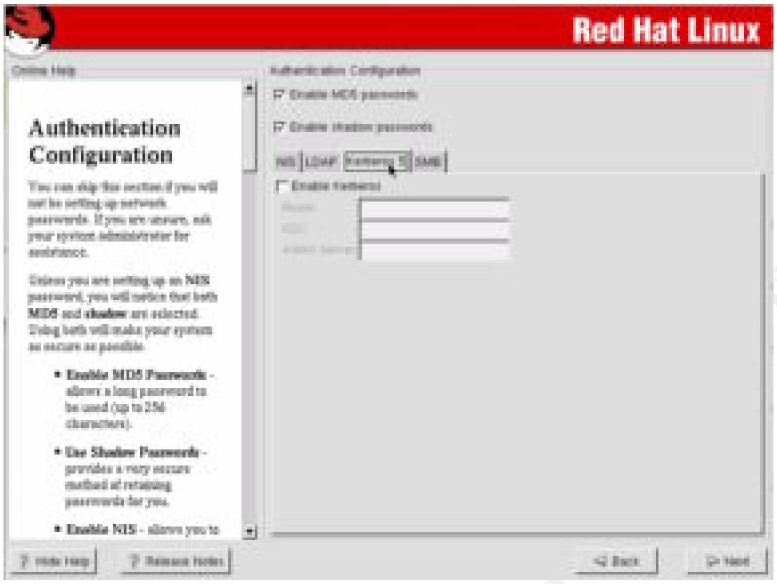
Figure 6 - LDAP Tab of Authentication Configuration Screen

Kerberos 5[12] Authentication

Kerberos 5 is a secure authentication and session encryption protocol. One of its advantages is the ability to authenticate users without sending passwords in clear text over the network. Red Hat Linux 7.3 can use an existing Kerberos 5 infrastructure to authenticate users onto the system. There are three aspects of a Kerberos authentication setup that must be configured for it to work. First, the Key Distribution Center (KDC) must be defined by IP address or name. The KDC distributes ticket-granting tickets (TGT) to successfully authenticated users that are members of a particular *realm*. The realm is the second piece of information that must be supplied to Red Hat Linux 7.3 for it to be used for authentication and is similar to the *domain* in NIS. It is a way to group clients together so that multiple sets of authentication data can be handled within one organization without them mixing. The IP address or name of the server running **kadmind** must also be specified. The **kadmind** handles administrative database functions like password changes[13]. Figure 7 is a screen shot of the Kerberos 5 tab of the Authentication Configuration screen.

[12] http://web.mit.edu/kerberos/www/

[13] http://web.mit.edu/kerberos/www/krb5-1.2/krb5-1.2.5/doc/admin.html



Authentication Configuration

You can skip this section if you will not be setting up network passwords. If you are unsure, ask your system administrator for assistance.

Unless you are setting up an NIS password, you will notice that both **MD5** and **shadow** are selected. Using both will make your system as secure as possible.

- **Enable MD5 Passwords** - allows a long password to be used (up to 256 characters).
- **Use Shadow Passwords** - provides a very secure method of retaining passwords for you.
- **Enable NIS** - allows you to

Authentication Configuration

☑ Enable MD5 passwords

☑ Enable shadow passwords

NIS | LDAP | Kerberos 5 | SMB

☐ Enable Kerberos

Realm:

KDC:

Admin Server:

Figure 7 - Kerberos 5 Tab of the Authentication Configuration Screen

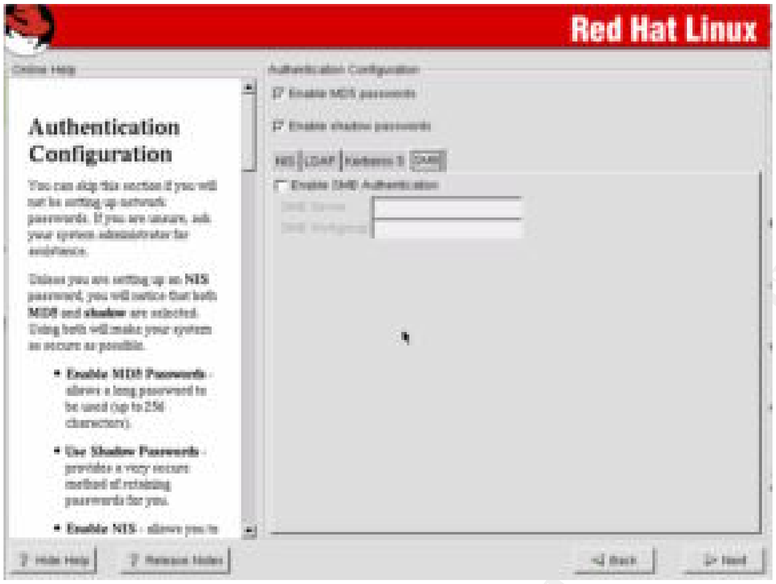
SMB Authentication

Server Message Block (SMB) protocol is that which is used in Windows networking. Red Hat Linux 7.3 can utilize a server that speaks SMB, like Windows XP Professional[14] or Samba[2], to authenticate a user. To set up the SMB authentication, two pieces of information are needed: the *workgroup* name and the server IP address or name. The workgroup is an organizational grouping similar to the *domain* in NIS and the *realm* in Kerberos 5.

Since SMB uses the ports 137 through 139, if the Firewall Configuration is set to Medium or High, SMB authentication will not work.

Figure 8 is a screen shot of the SMB tab of the Authentication Configuration screen.

[14] http://www.microsoft.com/windowsxp/pro/default.asp



Red Hat Linux

Online Help

Authentication Configuration

You can skip this section if you will not be setting up network passwords. If you are unsure, ask your system administrator for assistance.

Unless you are setting up an **NIS** password, you will notice that both **MD5** and **shadow** are selected. Using both will make your system as secure as possible.

- **Enable MD5 Passwords** - allows a long password to be used (up to 256 characters).
- **Use Shadow Passwords** - provides a very secure method of retaining passwords for you.
- **Enable NIS** - allows you to

Authentication Configuration

☑ Enable MD5 passwords

☑ Enable shadow passwords

NIS | LDAP | Kerberos 5 | SMB

☐ Enable SMB Authentication

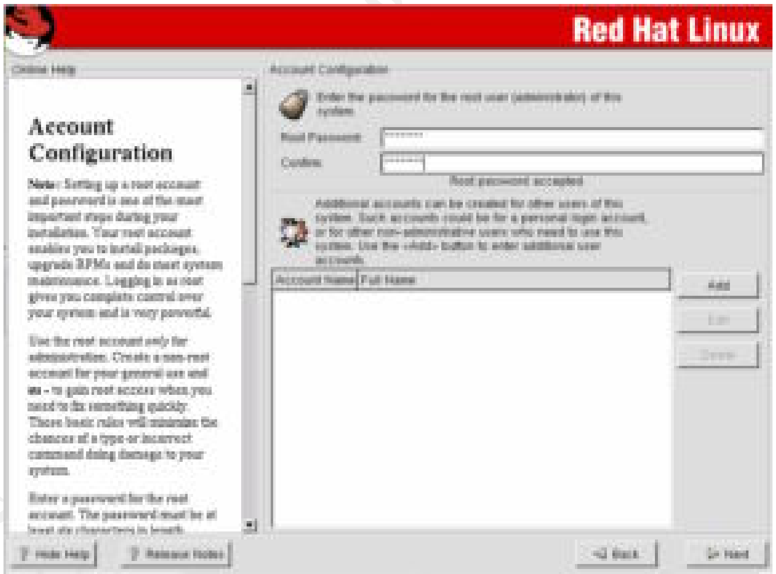
SMB Server

SMB Workgroup

Figure 8 - SMB Tab of the Authentication Configuration Screen

root Password

The installation program requests that a password for the *root* user account be entered.



Red Hat Linux

Online Help

Account Configuration

Note: Setting up a root account and password is one of the most important steps during your installation. Your root account enables you to install packages, upgrade RPMs and do most system maintenance. Logging in as root gives you complete control over your system and is very powerful.

Use the root account only for administration. Create a non-root account for your general use and **su** - to gain root access when you need to fix something quickly. These basic rules will minimize the chances of a typo or incorrect command doing damage to your system.

Enter a password for the root account. The password must be at least six characters in length.

Account Configuration

Enter the password for the root user (administrator) of this system.

Root Password: ********

Confirm: ********

Root password accepted

Additional accounts can be created for other users of this system. Such accounts could be for a personal login account, or for other non-administrative users who need to use this system. Use the <Add> button to enter additional user accounts.

Account Name	Full Name

Figure 9 - *root* Password Entry Screen

Since this account provides unrestricted access to the system, it is extremely important that one pick a strong password. Any account deserves a strong password for that matter. Strong passwords are hard to guess but easy to remember. Passwords should never contain words or names. Some examples of bad passwords are:

- fred375

- CarNut
- 1bonjovi

As a general rule of thumb, good passwords:

- Are at least 6 characters long
- Have a mixture of letter case, punctuation and numerals (e.g. 0 through 9)
- Have meaning for the user so they are easy to remember

More in-depth discussion on picking strong passwords can be found at http://consult.cern.ch/writeups/security/security_3.html.

If local accounts are to be created on the system, the *root* password entry screen provides a way to add them. If many local accounts are needed this method can become tedious and the standard **useradd** and **groupadd** tools can be used in conjunction with shell scripts to automate the task.

Post-Install Configuration

Patching

Every operating system produced has flaws, and new flaws are discovered in software components everyday. Red Hat is very good about quickly responding to and fixing bugs that are found in the many individual components, especially bugs that are security weaknesses. Carnegie Mellon University's CERT Coordination Center[15] and Incidents.org[16] are the leading web sources for up to date computer security vulnerability information. Red Hat seems to pay close attention to what these organizations find because they quickly provide patches for the problems.

Red Hat provides an Errata[17] page to look for any patches that are available for the most recent editions of their Linux distribution, including Version 7.3[18]. The patches can also be downloaded from Red Hat's FTP server[19].

Red Hat's patches are distributed in their RPM (Red Hat Package Management) formatted files just like their installation files. The **rpm** command is used to install the files. There are many parameters that the **rpm** command will accept. The ones that pertain to installation are the following:

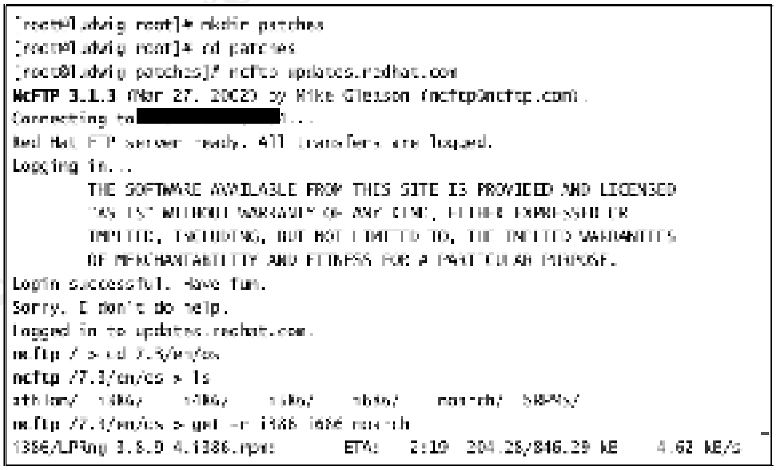
[15] http://www.cert.org
[16] http://www.incidents.org
[17] http://www.redhat.com/apps/support/errata/
[18] http://rhn.redhat.com/errata/rh73-errata.html
[19] ftp://updates.redhat.com

Table 1 - Some useful rpm **command-line options**

Parameter	*Purpose*	*Example*
-i	Installs the listed rpm files. Any rpm files that contain existing packages on the system will not be installed.	rpm –i junk.rpm
-U	If the packages in the listed rpm files exist, update them; otherwise, install them.	rpm –U junk.rpm
-F	Only updates packages that are already installed from the list of supplied rpm files. This is a good option for upgrading packages.	rpm –F junk.rpm
-e	Erases the packages whose names are listed on the command line.	rpm –e junk

For a more in-depth discussion on the use of the **rpm** command see http://www.redhat.com/docs/manuals/linux/RHL-7.3-Manual/custom-guide/ch-rpm.html. The **rpm** command performs an analysis of the dependencies that any installed packages and any packages in the listed rpm files may have. This keeps one from installing software that depends on something else that does not exist on the system.

The easiest way to update a newly installed system is to download all the new rpm files that are available from Red Hat's FTP site. The **ncftp** FTP client is handy for this because it can download whole directory structures. Only those packages intended for the computer's architecture should be downloaded. For example, only the **i386**, **i686** and **noarch** directories should be downloaded if the packages are to be installed on a Pentium 4 based computer. Figure 10 shows an example screen shot of an FTP session where only the patches needed in the above example are downloaded.

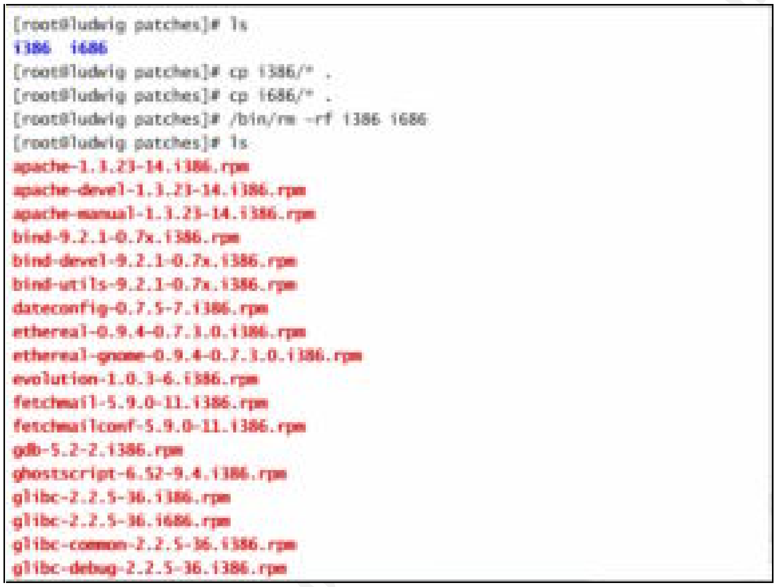


```
[root@ladvig root]# mkdir patches
[root@ladvig root]# cd patches
[root@ladvig patches]# ncftp updates.redhat.com
NcFTP 3.1.3 (Mar 27, 2002) by Mike Gleason (ncftp@ncftp.com).
Connecting to [redacted]...
Red Hat FTP server ready. All transfers are logged.
Logging in...
        THE SOFTWARE AVAILABLE FROM THIS SITE IS PROVIDED AND LICENSED
        "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR
        IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
        OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.
Login successful. Have fun.
Sorry, I don't do help.
Logged in to updates.redhat.com.
ncftp / > cd 7.3/en/os
ncftp /7.3/en/os > ls
athlon/  i386/    i486/    i586/    i686/    noarch/  SRPMS/
ncftp /7.3/en/os > get -R i386 i686 noarch
i386/LPRng-3.8.9-4.i386.rpm:      ETA:   2:19  204.28/846.29 kB    4.62 kB/s
```

Figure 10 - Example Download of Patches

First a directory is created to hold the downloaded files. The user then moves into the directory and uses the **ncftp** command to FTP to Red Hat's server for patches. After going into the directory that houses the rpm files for Version 7.3 (**7.3/en/os** - 'en' means English and 'os' means operating system), the user lists the contents of the directory. There are directories that contain rpm files for various processor architectures, but the user is only interested in those relevant to the Pentium 4. The **get –r** command is used to recursively download the contents of the desired directories.

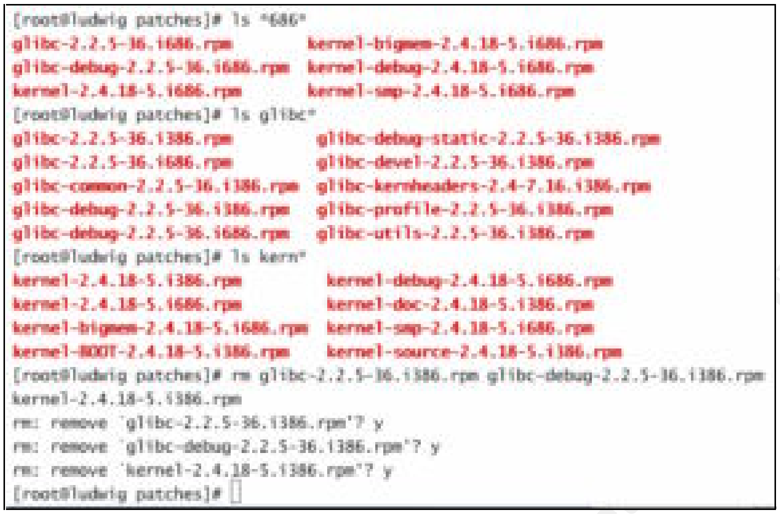
After the download completes, the patches are in the same directory structure that they were in on the FTP server, so to ease installing them it is best to move them to one directory. Figure 11 illustrates one way to accomplish this.



```
[root@ludwig patches]# ls
i386  i686
[root@ludwig patches]# cp i386/* .
[root@ludwig patches]# cp i686/* .
[root@ludwig patches]# /bin/rm -rf i386 i686
[root@ludwig patches]# ls
apache-1.3.23-14.i386.rpm
apache-devel-1.3.23-14.i386.rpm
apache-manual-1.3.23-14.i386.rpm
bind-9.2.1-0.7x.i386.rpm
bind-devel-9.2.1-0.7x.i386.rpm
bind-utils-9.2.1-0.7x.i386.rpm
dateconfig-0.7.5-7.i386.rpm
ethereal-0.9.4-0.7.3.0.i386.rpm
ethereal-gnome-0.9.4-0.7.3.0.i386.rpm
evolution-1.0.3-6.i386.rpm
fetchmail-5.9.0-11.i386.rpm
fetchmailconf-5.9.0-11.i386.rpm
gdb-5.2-2.i386.rpm
ghostscript-6.52-9.4.i386.rpm
glibc-2.2.5-36.i386.rpm
glibc-2.2.5-36.i686.rpm
glibc-common-2.2.5-36.i386.rpm
glibc-debug-2.2.5-36.i386.rpm
```

Figure 11 - Preparing Downloaded Patches for Install

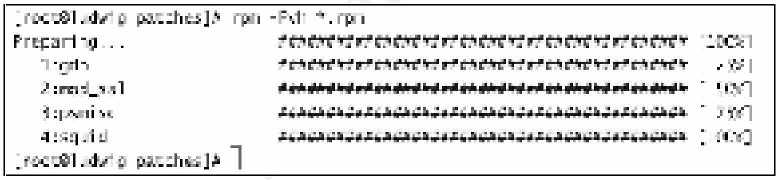
Once all the rpm files have been downloaded, any packages that are duplicates, except for the intended architecture, should be removed. For example, there will be a kernel package intended for the basic i386 based system and one for those that have i686 (Pentium Pro and newer) processors. The user must remove the redundant packages, or the **rpm** command will complain about dependencies. In the following illustration (see Figure 12), there are redundant **glibc** and **kernel** packages. Assuming an installation on a Pentium 4 computer, the packages for i386 should be removed so that the packages that are optimized for the Pentium 4 processor are used.



```
[root@ludwig patches]# ls *686*
glibc-2.2.5-36.i686.rpm        kernel-bigmem-2.4.18-5.i686.rpm
glibc-debug-2.2.5-36.i686.rpm  kernel-debug-2.4.18-5.i686.rpm
kernel-2.4.18-5.i686.rpm       kernel-smp-2.4.18-5.i686.rpm
[root@ludwig patches]# ls glibc*
glibc-2.2.5-36.i386.rpm         glibc-debug-static-2.2.5-36.i386.rpm
glibc-2.2.5-36.i686.rpm         glibc-devel-2.2.5-36.i386.rpm
glibc-common-2.2.5-36.i386.rpm  glibc-kernheaders-2.4-7.16.i386.rpm
glibc-debug-2.2.5-36.i386.rpm   glibc-profile-2.2.5-36.i386.rpm
glibc-debug-2.2.5-36.i686.rpm   glibc-utils-2.2.5-36.i386.rpm
[root@ludwig patches]# ls kern*
kernel-2.4.18-5.i386.rpm         kernel-debug-2.4.18-5.i686.rpm
kernel-2.4.18-5.i686.rpm         kernel-doc-2.4.18-5.i386.rpm
kernel-bigmem-2.4.18-5.i686.rpm  kernel-smp-2.4.18-5.i686.rpm
kernel-BOOT-2.4.18-5.i386.rpm    kernel-source-2.4.18-5.i386.rpm
[root@ludwig patches]# rm glibc-2.2.5-36.i386.rpm glibc-debug-2.2.5-36.i386.rpm
kernel-2.4.18-5.i386.rpm
rm: remove `glibc-2.2.5-36.i386.rpm'? y
rm: remove `glibc-debug-2.2.5-36.i386.rpm'? y
rm: remove `kernel-2.4.18-5.i386.rpm'? y
[root@ludwig patches]# 
```

Figure 12 - Removing Redundant Packages

Once the redundant rpm files are removed, the **rpm** command is used to install the packages. In most cases, the 'freshen' parameter to **rpm** can be used because one normally only wants to update what is already installed. Figure 13 shows an example session where **rpm –F** is used to update packages that are already installed.



```
[root@ludwig patches]# rpm -Fvh *.rpm
Preparing...                ########################################### [100%]
   1:gdm                    ########################################### [ 25%]
   2:mod_ssl                ########################################### [ 50%]
   3:psmisc                 ########################################### [ 75%]
   4:squid                  ########################################### [100%]
[root@ludwig patches]# 
```

Figure 13 - Using the rpm **command to install the patch rpm files**

The **v** and **h** parameters provide verbosity and hash marks so that the user has a better idea of the progress of the installation.

When **rpm** complains about dependencies, then the job of patching the system becomes a bit more difficult. In that case, one must look at Red Hat's Linux 7.3 Errata page for packages whose updates require packages that were not installed previously. Those new packages with their dependents should be installed first with the **–u** (update) or **–i** (install) parameter to square away the dependencies before running **rpm** with the **–F** (freshen) parameter.

Red Hat Network

To make the job of patching a Red Hat Linux 7.3 system even easier, Red Hat provides a for-pay service called the Red Hat Network[20]. When using this service the job of staying current with patches is completely automated. An icon on the panel alerts the user of the system when new patches are available and

[20] http://rhn.redhat.com/

walks the user through the process of installing them. For those organizations that can afford it, the Red Hat Network service is well worth the money.

Console Access Changes

The default installation of Red Hat Linux 7.3 provides the user that logs into the console (keyboard, mouse and screen directly attached to computer) with special privileges with respect to certain device files, the CTRL-ALT-DEL key sequence, and certain programs that control the functioning of the computer. The following sections will discuss each in more detail.

Device Permissions

By default, any user that logs into the console of a Red Hat Linux 7.3 computer has special ownership rights over various devices of the computer, such as the floppy drive. Depending on the situation, these permissions may be too lenient, and these special rights must be revoked.

The **/etc/security/console.perms** file defines what devices have their permissions changed and to what when a user logs into the console (or other terminals for that matter). It may be modified to suit just about any situation, from extreme lenience to no special rights at all. The file has three types of lines in it:

1. **Comments** – are lines that are blank or have a '#' character at the beginning. These lines exist for the benefit of the reader of the file only and are ignored by the system.
2. **Device Definitions** – are lines that assign a symbol to a set of device file names. For example,

   ```
   <scanner>=/dev/scanner /dev/usb/scanner*
   ```

   assigns the symbol **<scanner>** to the set of device files **/dev/scanner** and all device files that begin with **/dev/usb/scanner** (e.g. /dev/usb/scanner0).
3. **Permission Definitions** – are lines that describe what file permissions are assigned to a file when someone logs into the console and the permissions that are assigned when the console user logs out. For example,

   ```
   <console>  0600 <scanner>    0600 root
   ```

   tells the system to give the console user (only) read and write permissions to all the device files represented by the **<scanner>** symbol when they log in and give the *root* user (only) read and write permissions to the same set of device files when the console user logs out.

To have more restrictive permissions, all that is necessary to do is to remove entries from the /etc/security/console.perms file. The permission specification is the same that is used by the **chmod**[21] command, so the permissions assigned can be finely tuned to include group and world ownership.

Ctrl-Alt-Del *

By default, the CTRL-ALT-DEL (C-A-D) key sequence has special meaning in that it will cause the Red Hat Linux 7.3 computer to reboot. This may not be desirable in certain situations, and this function can be turned off. The **init** process is the ancestor of all the processes on the system and handles the capture of the C-A-D key combination and the processing of it. The configuration file for the **init** process is **/etc/inittab**. There is an entry in this file that looks like this:

```
ca::ctrlaltdel:/sbin/shutdown -t3 -r now          .
```

To remove the ability of all users, including *root*, to use C-A-D to reboot the computer, just put a '#' character in front of it to make it into a comment. To specify users by name that are allowed to use the C-A-D key combination to reboot the computer, create a file called **shutdown.allow** in the **/etc** directory[22] with each user listed on a separate line.

Special Program Access

The console user has permission to run various system operation programs that other remote users do not. There are files in the **/etc/security/console.apps** directory that define all the programs that can be run by the console user. Examples of special programs are **halt**, **shutdown**, and **reboot**. To remove the ability to run any of these special programs listed in the above noted directory, remove the file with the same name as the command that you want to disallow. Removing all the files in **/etc/security/console.apps** will remove the ability to run any of the programs as a console user[23].

Shutting Off Unneeded Services

It is important from a security standpoint to be running only the services that are necessary for providing the desired functions and no more. This cuts back on the avenues for attack. Therefore, it is important to know how to turn off unneeded functions.

Red Hat Linux 7.3 uses the AT&T UNIX concept of *run levels* to control the status of the system and to determine when to start and stop the programs that perform the functions of the computer. Run levels are numbered and each has a particular purpose. For example, run level 1 is for single user (administrative)

[21] **chmod** manual page - use 'man chmod' command to view

[22] http://www.redhat.com/docs/manuals/linux/RHL-7.3-Manual/custom-guide/ch-console-access.html

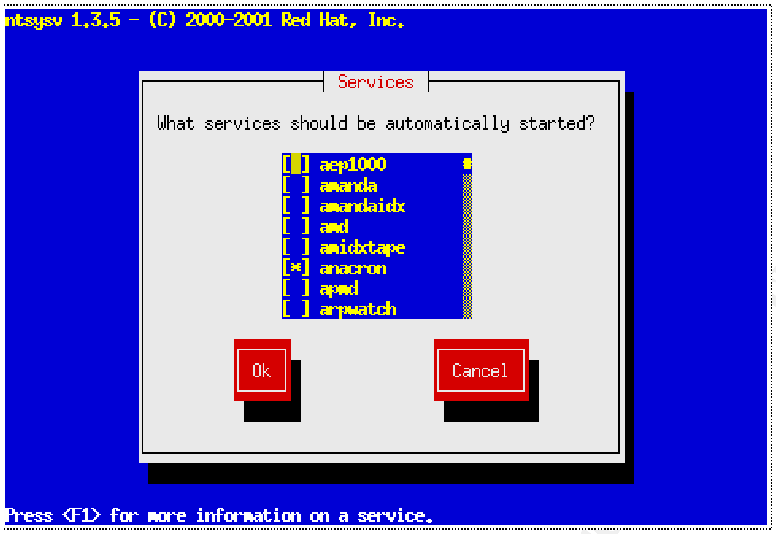
[23] http://www.redhat.com/docs/manuals/linux/RHL-7.3-Manual/custom-guide/s1-access-console-program.html

mode where only the basic system is running with one terminal (the console). Single user mode is for performing administrative duties, like file system repair, which require all other user activity to be off. A good discussion on the run level mechanism in Red Hat Linux 7.3 can be found at http://www.redhat.com/docs/manuals/linux/RHL-7.3-Manual/ref-guide/s1-boot-init-shutdown-booting.html.

All the scripts for starting and stopping services during boot (system starting up into run level 3 without X Windows or run level 5 with X Windows) and run level changes reside in the **/etc/init.d** directory. Each run level has a directory associated with it with links to the service scripts in **/etc/init.d**. For example, run level 3 has links to the appropriate start and stop scripts in **/etc/rc3.d**. The links are named in such a way to indicate whether the service is stopping or starting and order in which they should be run. Services that are to be stopped have links with names that start with 'K' (for kill). Services that are to be started have links with names that start with 'S'. After the letter that indicates starting or stopping, the name has a two-digit number that indicates the ordering by which it will run. The scripts are run in increasing order. For example, run level 3 starts the networking services with the **/etc/rc3.d/S10network** script by actually running the **/etc/init.d/network** script to which it is linked with the 'start' parameter. Similarly, links with 'K' names cause their associated scripts to be run with the 'stop' parameter.

Configuring services to be run or not run by hand at certain run levels is a tedious task, but Red Hat provides an easy-to-use tool, **ntsysv**, to assist in the task. **ntsysv** shows a list of all possible services with check boxes (see Figure 14). When a box is checked, then the corresponding service will start in the run level. By default, **ntsysv** configures services for the current run level, but run levels may be specified with the **--level <level(s)>** parameter. If more than one level number is listed (no spaces between the numbers), then the services checked or un-checked will be started or stopped, respectively, for all of the run levels listed with the **--level** parameter[24].

[24] **ntsysv** manual page -- use 'man ntsysv' command to view



```
ntsysv 1.3.5 - (C) 2000-2001 Red Hat, Inc.

                  Services

What services should be automatically started?

        [ ] aep1000
        [ ] amanda
        [ ] amandaidx
        [ ] amd
        [ ] amidxtape
        [*] anacron
        [ ] apmd
        [ ] arpwatch

Press <F1> for more information on a service.
```

Figure 14 - Example ntsysv Screen

The **ntsysv** utility actually uses a more basic non-interactive command **/sbin/chkconfig**. This command and, therefore, **ntsysv** use specially formatted comments in the service scripts to know in which run levels a service should start and stop and where in the order they should be started and stopped[25]. For example, the **/etc/init.d/network** script has the comment lines in it:

```
# chkconfig: 2345 10 90
# description: Activates/Deactivates all network interfaces configured to \
#              start at boot time.
```

The first line tells **/sbin/chkconfig** that:

- The service starts in run levels 2, 3, 4, and 5
- The service stops in run levels 0, 1, and 6 (because they are not listed in the 'start' list)
- The service is given a number 10 (close to the beginning) in the start ordering and a 90 (close to the end) in the stop ordering.

The **/sbin/chkconfig** command, in addition to controlling the services handled by the run level scripts, controls the services provided by the **xinetd** "super server", which is discussed in more detail in the **Restricting Access to Services with TCP Wrappers** section below.

Restricting Access to Services with TCP Wrappers

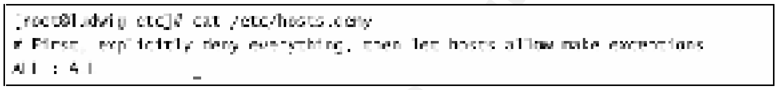
The **xinetd** "super server" process handles many services that a Red Hat Linux 7.3 computer can provide to clients on the network, like FTP access. **xinetd** is configured via the **/etc/xinetd.conf** file and the individual files in the **/etc/xinetd.d** directory. Normally, these services have no way to restrict which

[25] **chkconfig** manual page -- use 'man chkconfig' command to view

systems on the network are allowed to connect to them[26], or if they do, they have different mechanisms for doing so which adds to the administrative confusion.

TCP Wrappers is a software mechanism that homogenizes the way that access is restricted to network services. Wietse Venema wrote the software when he was at Eindhoven University in the Netherlands[27]. Software, like **xinetd**, that has TCP Wrappers incorporated into it uses two files for setting access restrictions, **/etc/hosts.allow** and **/etc/hosts.deny**. As their names denote, **/etc/hosts.allow** contains settings for allowing access, and **/etc/hosts.deny** contains settings for denying access. TCP Wrappers allows access control on IP addresses and names and user names, if the **ident** service is running on the requestor[28].

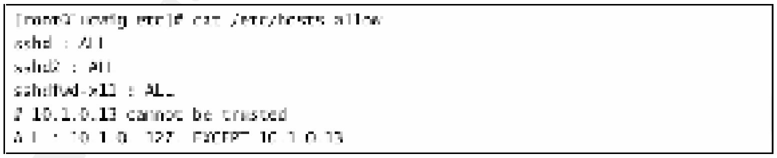
The syntax for the files is the same for each, and an example is the best way to learn it. Figure 15 illustrates a listing of a basic **/etc/hosts.deny** file. The first line is a comment (starts with the '#' character) and is ignored during processing. The second line is the only directive, and it means that requests for all services (the first **ALL**) are denied from all computers (the second **ALL**).



```
[root@ladvig etc]# cat /etc/hosts.deny
# First, explicitly deny everything, then let hosts.allow make exceptions
ALL : ALL
```

Figure 15 - /etc/hosts.deny Example

The **/etc/hosts.allow** file defines exceptions to the rules that are provided in **/etc/hosts.deny**. Figure 16 shows an example **/etc/hosts.allow** file. The first three lines allow all hosts (the second **ALL** in each line) access to the **sshd**, **sshd2**, and **sshdfwd-x11** services. The fourth line is a comment. The last line grants access to all services (the first **ALL** in the line) from hosts in the 10.1.0.0/24 network and the 127.0.0.0/8 network (the local host itself), except for the 10.1.0.13 host.



```
[root@ladvig etc]# cat /etc/hosts.allow
sshd : ALL
sshd2 : ALL
sshdfwd-x11 : ALL
# 10.1.0.13 cannot be trusted
ALL : 10.1.0. 127. EXCEPT 10.1.0.13
```

Figure 16 - /etc/hosts.allow Example

TCP Wrappers directives can be much more elaborate. In general, the directive syntax is as follows:

[26] http://www.redhat.com/docs/manuals/linux/RHL-7.3-Manual/ref-guide/ch-tcpwrappers.html
[27] http://www.porcupine.org/wietse/
[28] ftp://ftp.porcupine.org/pub/security/index.html

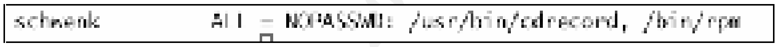
```
service list : host list [ : shell commands ]
```
[29]

The *shell command* after the second colon can be any program or shell script that the administrator wishes. This provides infinite possibilities for system monitoring and simple intrusion attempt detection. For example, the administrator can use a shell command to email the pertinent information when a particularly interesting connection attempt is made.

Delegating Administrative Duties with Sudo

Normally, logging into a Red Hat Linux system as the *root* user is required to perform administrative functions. There is a program distributed with Red Hat Linux 7.3, called **sudo** that can be configured by the *root* user to allow specific users the ability to run certain programs as if they were the *root* user. This can be handy for delegating administrative duties to trusted users.

The configuration file, **/etc/sudoers**, is somewhat complex. A special program, **visudo**, is provided to automate the editing and error-checking of the configuration file. Reading the manual page[30] for it is required reading for competent configuration of **sudo**. This section will provide a simple example to demonstrate the power of the **sudo** utility. As usual, blank lines and comments (lines beginning with '#') are ignored.



```
schwenk         ALL = NOPASSWD: /usr/bin/cdrecord, /bin/rpm
```

Figure 17 - Example Line from /etc/sudoers

Figure 17 illustrates an example **/etc/sudoers** directive. In it, the user 'schwenk' on 'ALL' hosts is allowed to run both **/usr/bin/cdrecord** and **/bin/rpm** without the need to type in his password ('NOPASSWD:'). If the program names are not specified with the exact path, then any program with a matching name will be allowed. This may cause an undesired elevation of privileges. Therefore, it is best to be specific about which instance of a program is meant.

Upkeep and Monitoring

After the initial installation and configuration of a Red Hat Linux 7.3 computer, it is important to perform frequent and ongoing maintenance and monitoring of the system. The following sections will discuss the basic duties that should be performed to ensure that the computer remains as securely configured as possible.

Patch Management

As discussed in the **Patching** section on page 13, security-related problems are constantly being found in the software components shipped with Red Hat Linux

[29] **hosts_access** manual page in section 5 -- use the 'man 5 hosts_access' command to view

[30] **sudoers** manual page -- use the 'man sudoers' command to view

7.3. The initial patching of the system should be followed up with regular bouts of patch installations. How frequently one should patch the system depends on many factors, including (but not limited to):

- Organizational security policy
- Balance between security and administrative workload
- The importance of the services performed by the system (mission-critical server vs. infrequently-used word-processing station)

The easiest way to keep a system current with regard to software updates is to subscribe to the Red Hat Network[20], but this for-pay service may not be affordable by all organizations. Red Hat provides an email list, **redhat-watch-list**[31], which they use to announce all the new patches. This service is invaluable even for those who subscribe to the Red Hat Network so that one can stay abreast of newly found problems in the Red Hat Linux 7.3 product. When an update is announced, all the rpm files associated with it are listed for each supported version of Red Hat Linux. This eases the job of knowing what to download for a particular fix. Also, using the mailing list gives the administrator the opportunity to decide whether or not an update is warranted for the particular situation.

As a final point on patching a Red Hat Linux 7.3 system, it is beneficial to download all the patches into a location that is accessible throughout the organization, like a file server, or on removable media, like a CD. This repository can be updated frequently as new patches are announced on the mailing list, so that newly installed systems can be updated with the latest packages with little difficulty.

Monitoring

Much of keeping a computer system secure involves monitoring the system for breaches. No system is completely secure, so it is important to monitor them for the possibility of intrusion and other exploits. This section will briefly discuss various important aspects of monitoring for security problems in a Red Hat Linux 7.3 system, such as log monitoring, intrusion detection, and file alteration detection.

Event Logging

Most of the software provided with Red Hat Linux 7.3 is configured to utilize the built-in logging facility, **syslog**. The **syslog** configuration file **/etc/syslogd** instructs **syslogd** (the logging server process) to store most of the entries in files in the **/var/log** directory and sub-directories thereof. It is relatively simple to determine where the log files are for the particular service or software of interest just by perusing the names of the files. For example, Samba puts its log files in **/var/log/samba**.

[31] http://www.redhat.com/mailing-lists/

Weeding through the log files for entries of interest can be tedious and difficult, but Red Hat Linux 7.3 provides a handy utility, **logwatch**, to automate the task a bit. **logwatch**, among other things, can be instructed to search the log files for entries that meet particular criteria and generate a report to screen. By default, Red Hat has a **cron**[32] job configured to run **logwatch** on a daily basis so that it emails the *root* user with a report of security and system function events of interest. For example, the **logwatch** report contains entries for **sshd**[33] activity, so that the administrator can monitor for trouble. **logwatch** is very configurable. A detailed description can be found in the manual page for it[34].

File Alteration Detection with Tripwire

Red Hat provides a preconfigured installation of the open-source version of the Tripwire[35] file alteration detection software. Full instructions for configuring it can be found at http://www.redhat.com/docs/manuals/linux/RHL-7.3-Manual/ref-guide/ch-tripwire.html, but essentially the steps are:

1. Make sure the software is installed. The **rpm** program in 'query' mode can do this. For example,
   ```
   rpm –q tripwire
   ```
   will output the complete package name (e.g. 'tripwire-2.3.1-10') if it is installed already and 'package tripwire is not installed' if it is not.
2. Customize the default configuration file, **/etc/tripwire/twcfg.txt**, and run the configuration script, **/etc/tripwire/twinstall.sh**.
3. Build the initial Tripwire database using the
   ```
   /usr/sbin/tripwire --init
   ```
   command.
4. Do an integrity check to see if all the monitored files are the desired ones by running the
   ```
   /usr/sbin/tripwire --check
   ```
   command. If changes need to be made, return to step 2 and fix the problem.

At this point, Tripwire is configured. Red Hat provides a **cron** job in the default installation of Red Hat Linux 7.3 that runs an integrity check on a daily basis, so all the administrator needs to do is monitor his or her email to stay informed of any file integrity problems that would indicate a breach.

Stay Informed

The information security field is in a state of constant flux. There are many standard security practices, with which everyone should be familiar, but new holes are being found constantly, and one should stay on top of the ever-changing information security field. This section will discuss various sources of

[32] **cron** is a program that is common in UNIX environments that is used to schedule the running of programs on a repetitive schedule

[33] Secure Shell – see http://www.openssh.org

[34] **logwatch** manual page – use 'man logwatch' command to view

[35] http://www.tripwire.org

information to aide in the task of staying informed, like web sites, mailing lists and professional certifications.

Web Sites

There are many web sites devoted to information security topics, not necessarily Red Hat Linux specific, but they are good to be aware of in any event. The following is a non-exhaustive list of some good ones.

- http://www.cert.org - Carnegie Mellon University's CERT Organization
- http://www.incidents.org - SANS's Incidents.org Site
- http://www.linuxsecurity.com/ - Linux-specific information
- http://www.cerias.purdue.edu/ - Center for Education and Research in Information Assurance and Security at Perdue University
- http://www.alw.nih.gov/Security/security-www.html - a good catch-all list of sites

Specific information about information security can be extracted from the Web using one of the many search engines available, like http://www.google.com.

Mailing Lists

In addition to actively searching for information security information, it can be sent directly to the interested administrator via the many security related mailing lists. The following is a list of a few good information security mailing lists and methods for signing up for them.

- redhat-watch-list@redhat.com - see https://listman.redhat.com/
- SANS Security Alert Consensus - see http://www.sans.org/sansnews/
- cert-advisory@cert.org - see http://www.cert.org/contact_cert/certmaillist.html

Training

Many organizations can provide training in information security. Training can be an important facet of an administrator's overall security strategy because it can provide a more intense learning experience compared with unfocused wading through on-line references and books. The following is a brief list of some good sources for security training.

- Red Hat provides the RHCE (Red Hat Certified Engineer) certification provides Linux-centric instruction on systems security - see http://www.redhat.com/training/
- Carnegie Mellon University's CERT organization - see http://www.cert.org/training/
- MIS Training Institute - see http://www.misti.com
- SANS's GIAC (Global Information Assurance Certification) organization focuses solely on information security training and certification – see http://www.giac.org

Conclusion

This article provided guidelines for securing a Red Hat Linux 7.3 computer. It covered the pre-installation, installation and upkeep of such a system. Securing computer systems is an ongoing process that requires many resources to do it well, and this article provided a starting point in the progression. The reader must determine how this information fits within his or her organization's security policy and apply it appropriately.

References

Ballard, Josh. "IPTables vs. IPChains Comparison." URL: http://www.oofle.com/iptables/comparison.htm (25 July 2002).

Carnegie Mellon University. URL: http://www.cert.org (25 July 2002).

Kiley, Joe. "Getting Around BIOS Passwords." ZephirTech. March 2001. URL: http://www.zephirtech.com/misc/biospasswords.html (25 July 2002).

Linux IP Firewalling Chains. URL: http://netfilter.samba.org/ipchains (25 July 2002).

Manual pages for **chmod(1)** command, **hosts_access(5)**, **logwatch(8)** package and **sudoers(5)** file. Number in parentheses is manual section.

Massachusetts Institute of Technology. "Kerberos: The Network Authentication Protocol." URL: http://web.mit.edu/kerberos/www/ (25 July 2002).

Massachusetts Institute of Technology. "Kerberos V5 System Administrator's Guide - Version 1.2." 1.0. 16 June 2000. URL: http://web.mit.edu/kerberos/www/krb5-1.2/krb5-1.2.5/doc/admin.html (25 July 2002).

Microsoft Corporation. "Windows XP Professional Home Page." 19 July 2002. URL: http://www.microsoft.com/windowsxp/pro/default.asp (25 July 2002).

Netfilter. URL: http://netfilter.samba.org (25 July 2002).

OpenLDAP. URL: http://www.openldap.org/ (25 July 2002).

OpenLDAP. "Introduction to OpenLDAP Directory Services." OpenLDAP 2.1 Administrator's Guide. URL: http://www.openldap.org/doc/admin21/intro.html (25 July 2002)

OpenBSD. "OpenSSH" 23 July 2002. URL: http://www.openssh.org/

Red Hat, Inc. "Errata: Security Alerts, Bugfixes, and Enhancements" URL: http://www.redhat.com/apps/support/errata/ (25 July 2002).

Red Hat, Inc. "Red Hat Linux 7.3 - The Official Red Hat Linux x86 Installation Guide" URL: http://www.redhat.com/docs/manuals/linux/RHL-7.3-Manual/install-guide/ (25 July 2002).

Red Hat, Inc. "Red Hat Linux 7.3 - The Official Red Hat Linux Customization Guide." URL: http://www.redhat.com/docs/manuals/linux/RHL-7.3-Manual/custom-guide/ (25 July 2002).

Red Hat, Inc. "Red Hat Linux 7.3 – The Official Red Hat Linux Reference Guide." URL: http://www.redhat.com/docs/manuals/linux/RHL-7.3-Manual/ref-guide/ (25 July 2002).

Red Hat, Inc. "Red Hat Linux 7.3 General Advisories." URL: http://rhn.redhat.com/errata/rh73-errata.html (25 July 2002)

Red Hat, Inc. "Red Hat Mailing List Archives." URL: http://www.redhat.com/mailing-lists/ (25 July 2002).

Red Hat, Inc. "Red Hat Network." URL: http://rhn.redhat.com/ (25 July 2002).

Red Hat, Inc. Red Hat Updates FTP Server. URL: ftp://updates.redhat.com

Russell, Rusty. "Linux IPCHAINS-HOWTO." v1.0.8, 4 July 2000. URL: http://www.tldp.org/HOWTO/IPCHAINS-HOWTO.html (25 July 2002).

Samba. "SAMBA – opening windows to a wider world." URL: http://us1.samba.org/samba/samba.html (25 July 2002).

SANS Institute. "Incidents.org – The Internet Threat Monitor." URL: http://www.incidents.org (25 July 2002).

Tripwire, Inc. "Tripwire.org – Home of the Open Source project." URL: http://www.tripwire.org/ (25 July 2002).

Venema, Wietse. "Wietse's tools and papers." URL: ftp://ftp.porcupine.org/pub/security/index.html (25 July 2002).

Venema, Wietse. "Wietse Zweitze Venema." URL: http://www.porcupine.org/wietse/ (25 July 2002).