Global Information Assurance Certification Paper

Step-by-step Guide to

Securing

Red Hat 7.1 Linux

Lawrence D. Grim, Jr.
GCUX Practical Exercise
Version 1.6d

THIS PAGE LEFT BLANK INTENTIONALLY

Table of Contents

Disclaimer 1
Typing Conventions 1
Purpose of Exercise 2
Equipment 2
Installation 3
  Pre-installation setup 3
  Install using RedHat 7.1 Graphical Interface. 4
  Post installation comments 8
Hardening the system 8
  The INIT changes 8
  Sticky Bits 9
  SERVICES changes 9
  Installation of WU-FTPD 10
  The tcp_wrapper Process 10
  Banner changes 10
  SETUID/SETGID File Check 11
  World-writable Directories 11
  Password Security 11
  Timeout for Inactivity 12
  Limit the history of shell commands 12
  Limiting root access 12
  Setting the root UMASK 12
  Shutdown Unwanted Services 12
  Limit SuperUser capability 13
  OpenSSH security 13
  Installation of Additional Packages for Web Server 14
  Generating an SSL Key 14
  Using Apache Configuration Tool 15
  Apache WebServer Configuration 15
  Starting the Apache Web Server 15
  Updating the System 15
Conclusion 16

THIS PAGE LEFT BLANK INTENTIONALLY

Disclaimer

Use the information in this document at your own risk. I disavow any potential liability of this document. Use of the concepts, examples, and/or other content of this document is entirely at your own risk. This guide is written in the hope that it will be useful, but without any warranty; without even the implied warranty of merchantability or fitness for a particular purpose.

All copyrights are owned by their owners, unless specifically noted otherwise. Third party trademarks or brand names are the property of their owners. Use of a term in this document should not be regarded as affecting the validity of any trademark or service mark.

Naming of particular products or brands should not be seen as endorsements.

Typing Conventions

In this document, several actions are prompted or required. To avoid confusion, all on-screen prompts are in bold Courier typeface, like this **[root@GRIMRH /]#** Characters that are typed in are in regular Courier typeface, like this `kill -HUH 488`. Special keys, such as the enter key, will be put in italic Courier typeface, such as *ENTER* or *SHIFT*.

When editing files, normally the standard vi editor is used. The manual page for the vi editor is provided at a web site noted in the bibliography. Almost all UNIX systems have the vi editor, so practice with this editor is a basic skill.

Purpose of Exercise

Installation and securing a current Linux operating system on a server. The server will have standard, good security practices applied.

Equipment

Chosen for the system was a Dell OptiPlex GX100. This Intel Celeron 566MHz system has a 66MHz bus and 128K cache with 128 megabytes (MB) on one synchronous dynamic random-access memory (SDRAM) dual in-line memory module (DIMMs) in one of two DIMM sockets on the system board. The motherboard supports up to 512MB of SDRAM DIMM memory.

The motherboard is based on the Intel 810 system chip set with a data bus width of 64 bits and an address bus width of 32bits. The system clock runs at either 66 or 100MHz, matching the external bus speed.

It has an integrated Intel Direct Accelerated Graphics Port (AGP) 2D and 3D graphics accelerator with a display cache of 4 MB, 100-MHz synchronous dynamic random-access memory (SDRAM). The AGP video supports the following settings:

800 x 600 pixels; 85 hertz (Hz) refresh rate with 16.7 million colors
1024 x 786 pixels; 85 Hz refresh rate with 64,000 colors
1280 x 1024 pixels; 85 Hz refresh rate with 256 colors
1600 x 1200 pixels; 75 Hz refresh rate with 256 colors.

Figure 1: Video Settings

Included is Universal Serial Bus (USB) capability, which can simplify connecting peripheral devices such as mice, printers, and computer speakers. The system basic input/output system (BIOS) provides support for USB keyboards and mice in MS-DOS and other non-Windows environments. To use this capability, the USB support, it must be enable in the BIOS setup.

The hard disk controller provides Ultra Advanced Technology Attachment (ATA)/66 support, which allows storage devices to transfer data at speeds up to 66 MB per second (MB/sec). Separately installed is a Maxtor 531DX Ultra DMA 100 5400 RPM single head/single platter hard drive with an Ultra ATA/100 interface, with 29,297 actual cylinders, 16 heads and 63 sectors for a 15.0GB drive capacity. For software installation, a Memorex CD48-2E 48x speed CDROM with a internal ATAPI EIDE interface and a 128K Built-in Cache Buffer memory were mounted temporarily on the second ATA/66 IDE motherboard connection.

The imbedded Network Interface Controller (NIC) is a 3Com 3C900B TPC, 3C905C-TXM 10/100 Ethernet Adapter.

The external bus has two Peripheral Component Interconnect (PCI) slots in a small form-factor chassis at a bus speed of 33Mhz.

One external PCI slot contains an add-on Creative Labs SoundBlaster 16 PCI card (CT4740) using the CT5880 Audio PC chip which provides external connections for line and microphone inputs, line level and speaker outputs, as well as Musical Instrument Digital Interface (MIDI)/Joystick DB15 connector. The internal connectors support internal telephone answering device (TAD)/modem connection, a CDROM drive audio connection, and video television/MPC-3 analog CDROM audio connection.

External connections include:

a. two 9-pin connector 16550-compatable serial (data terminal equipment DTE) ports,
b. one DB25 bi-directional parallel port, a DB15 VGA connection,
c. one RJ45 integrated network interface (NIC) connector,
d. a pair of 6-pin mini-Deutsche Industrie Norm (DIN) IBM Personal System/2 (PS/2)-style keyboard and mouse connections and
e. two Universal Serial Bus (USB) compliant connectors.

During the power-on self-test (POST), following key combinations are available:

<Ctrl><Alt><Del>	restarts (reboots) the system
<Ctrl><Alt><\>	toggles microprocessor speeds on 101-key keyboard (in MS-DOS® real mode only)
<Ctrl><Alt><#>	toggles microprocessor speeds on 102-key keyboard (in MS-DOS real mode only)
<F2> or <Ctrl><Alt><Enter>	starts embedded System Setup (during power-on self-test [POST] only)
<F3> or <F12>	automatically starts (boots) the system from the network environment specified by the managed boot agent (MBA) rather than from one of the devices in the System Setup Boot Sequence option
<Ctrl><Alt><F10>	launches the utility partition (if installed) during system start-up

Figure 2: POST Keyboard Settings

Installation

Pre-installation setup

1. Using the *F2* key during the POST-test screen, open up the BIOS configuration setup.
2. Insure that the hardware installed (hard drive and CDROM) are recognized.
3. In the Boot Sequence selection, move the IDE CD-ROM device to the first boot device, with the Hard-Disk Drive C: as the second device.
4. Disconnect the computer from the network before the installation. After several hardening techniques are used, it can be brought online.

Install using RedHat 7.1 Graphical Interface.

1. When the text screen "**Welcome to Red Hat Linux 7.1!**" comes up, just hit the *ENTER* key. There will be a minute or so pause, while the bottom of the screen says **Running anaconda - Please wait.** Then, the graphical interface will start.
2. Language Selection: English
3. Keyboard Configuration: 102-key (International) PC model, U.S. English Layout with Enable Dead Keys.
4. Mouse Configuration: Generic 2 Button Mouse (PS/2), with Emulate 3 buttons box checked.
5. The screen **Welcome to Red Hat Linux** is just an informational screen, so press *NEXT*.
6. Installation Type: Custom System
7. Disk Partitioning: Manual partition with fdisk (experts only)
8. Select drive to partition, select the hda drive.
9. Using Fdisk: A simulated terminal screen opens. The prompt of "**Command (m for help):**" is provided after each step is accomplished.
   a. Enter `n` (for new), then press the *ENTER* key. Then, answer `p` (for primary), followed by the *ENTER* key. The Partition number will be 1 (for the first partition). (Press *ENTER* to continue) The first cylinder will be the 1, the default, accepted by *ENTER*. The last cylinder will be noted by `+3500M` for the 3.5GB allocated to the first partition.
   b. Follow the same procedures for the next partition, using the start/stop cylinder count shown in figure 3 below.
   c. Change the type of the second partition to a Linux swap partition by entering t at the prompt "**Command (m for help):**". Select the partition 3, and the Hex code to 82. Note that partition number 4 is an EXTENDED partition, not a primary partition.
   d. Check the allocation with the "`p`" command. It should look like:

```
Disk /tmp/hda: 255 heads, 63 sectors, 1823 cylinders
Units = cylinders of 16065 * 512 bytes

Device Boot   Start End       Blocks    Id    System
/tmp/hda1         1   447      3590496  83    Linux
/tmp/hda2       448   513       530145  83    Linux
/tmp/hda3       514   577       514080  83    Linux swap
/tmp/hda4       578  1823     10008495   5    Extended
/tmp/hda5       578   960      3076416  83    Linux
/tmp/hda6       961  1088      1028128  83    Linux
/tmp/hda7      1089  1216      1028128  83    Linux
/tmp/hda8      1217  1344      1028128  83    Linux
/tmp/hda9      1345  1599      2048256  83    Linux
/tmp/hda10     1600  1823      1798248  83    Linux
```

Figure 3: fdisk partition allocation

   e. Write the new partition sizes to the drive with `w` at the "**Command (m for help):**" prompt, which will also exit the fdisk utility.

f. Move off the graphical screen for using fdisk with the *NEXT* graphical button at the bottom of the screen.
g. The reason for the many separate partitions being setup during the installation was to use separate file system access controls on a partition. Generally, the files in the `/usr` mount point are tampered with by a system cracker or malicious user. The `/tmp`, `/var/spool` and `/var/log` mount points are restricted in size because some Denial of Service (DoS) attacks fill these areas with temporary files or log messages. If these are part of the `/` or `root` partition, such a DoS attack would fill up the root partition and cause a crash.

10. The Partitions screen will list the nine available partitions. Note that the extended drive hda4 is not available.
11. Select each partition, starting with the first device hda, using the mouse to highlight the entry. After highlighting the device, use the Edit graphical button and change the Mount Point for each partition to the one listed below:

Mount point	Device	Requested	Actual	Type
/	hda1	3506M	3506M	Linux native
/var	hda2	517M	517M	Linux native
<Swap>	hda3	502M	502M	Linux swap
/usr	hda5	3004M	3004M	Linux native
/opt	hda6	1004M	1004M	Linux native
/var/spool	hda7	1004M	1004M	Linux native
/var/log	hda8	1004M	1004M	Linux native
/tmp	hda9	2000M	2000M	Linux native
/home	hda10	1757M	1757M	Linux native

Figure 4: Mount Points for Partitions

12. Go to the next screen with the graphical *NEXT* button.

13. Making sure that "Check for bad blocks while formatting" is selected, leave all of the partitions automatically selected for formatting. Make sure to check the **`/dev/hda10 /home`** block also. Use the *NEXT* graphical button to proceed to the next screen.

14. LILO Configuration will be filled in with the defaults. Because this system did not have a floppy drive, unselect the **`Create boot disk`** option at the top of the screen. For convenience, all other selections were left at the default. Select *NEXT*.

15. Under Network configuration, unselecting the box for **`Configure using DHCP`** opened up the lower selections. Note that we left the "Activate on boot" checked/selected. For our system, we used the following information in the lower portion:

IP Address	192.168.100.251

Netmask	255.255.255.224
Network	192.168.100.224
Broadcast	192.168.100.255
Hostname	GRIMRH.GRIM.COM
Gateway	192.168.100.228
Primary DNS	192.168.96.20
Secondary DNS	192.168.80.33
Ternary DNS	192.168.20.14

Figure 5: Network Configuration

Press *NEXT* to move to the Firewall screen.

16. As an option, we selected to have no firewall for the security level at the Firewall Configuration screen. (*NEXT*)
17. For Language Support, we left the default of English (USA) selected. (*NEXT*)
18. For Time Zone selection, we were able to leave the selected America/New York or Eastern Time choice. For the **UTC offset** tab, we verified that choice of **UTC-05 US Eastern**. At the bottom of the **UTC offset** tab, we provided support for Daylight Savings Time by checking that box before proceeding to the next screen.
19. Under Account Configuration, we entered the root password (8 letter/numbers) twice, and created a user called ldgrim, with the same password, and full name of Larry Grim. Use the **Add** graphical button to move that entry into the bottom Account Name box. Later, this will be the sole user that can change to root with su. Another user, secldg, with the name of Larry Grim, will be our normal user. Use the **Add** button to move it to the bottom, then *NEXT*.
20. The next screen for Authentication Configuration was left with the default selections to enable MD5 and shadow passwords (*NEXT*)
21. Selection of the package groups allowed us to select Printer Support, X Window System, KDE, Mail/WWW/News tools, DOS/Windows Connectivity, Games, Multimedia Support, Web Server, Network Management Workstation, Development, Kernel Development, and Utilities. We unselected the GNOME and Dialup Workstation. This selection noted a total installation size of 1,135M at the lower right of the screen.
22. Because we selected the X Window System, the X Configuration screen came up next. The installation process correctly identified the Intel 810 video chipset. But we had to have the video card RAM changed from the detected 16MB to 4MB, using the graphic tab at the bottom of the screen. Next gave us the monitor selection, which was changed to the end of the listing as the un-probed monitor with horizontal sync range from 31.5 – 48.5 kHz, and vertical sync from 50-70 Hz. The next graphical screen gave us a default of High Color (16 bit) color depth, and a screen resolution of 1024x768. As a preference, we changed the screen resolution for our 15 inch monitor to 800x600. We also changed from the login type of graphical to text. This would make diagnosis of any X-window problems easier. (*NEXT*)

23. With all of the selections made, the formatting of the drives, installation of software, and post-installation steps were started with the Installation screen. About 12 minutes into the software installation, the screen will prompt to have the second RedHat cdrom inserted after it “spits out” the first cdrom.
24. After the reboot, the system will come up in text mode, with the prompt **`GRIMRH login:`**

Post installation comments

1. Prior to any hardening techniques, for information, the system looks like this:

Mount point	Device	Actual	Used	Percentage	Type
/	hda1	3506M	63792	2%	Linux native
<Swap>	hda2	517M			Linux swap
/var	hda3	502M	14384	4%	Linux native
/usr	hda5	3004M	983924	35%	Linux native
/opt	hda6	1004M	20	1%	Linux native
/var/spool	hda7	1004M	60	1%	Linux native
/var/log	hda8	1004M	224	1%	Linux native
/tmp	hda9	2000M	472	1%	Linux native
/home	hda10	1757M	148	1%	Linux native

Figure 6: Disk Usage

2. Shutdown the system, using the `shutdown -h now` command.
3. Disconnect the CDROM and close the case. When the system is restarted, enter the BIOS setup with the *F2* key.
4. Remove the second IDE device, now labeled as **Unknown** and insure that the boot sequence only has the IDE C-drive
5. When restarting the RedHat software, the initialization routine recognizes the change in hardware and prompts you to accept the changed system configuration.

Hardening the system

The INIT changes

1. Disable CTRL+ALT+DELETE reboots
   a. After the system has rebooted, log in as the alternative administrative user "ldgrim" Use the `su` command to elevate to the super-user or root privileges.
   b. Change directory to /etc with the `cd /etc/` command at the **GRIMRH** prompt.
   c. Change the file permissions on the inittab file with the command `chmod 744 inittab`
   d. Open the inittab file in the visual editor, with the command `vi inittab`.
   e. Move the cursor, with the arrow keys, down to the shutdown line.
   f. Hit the *INSERT* key at the beginning of the line and change it to read:
   `#ca::ctrlaltdel:/sbin/shutdown -t3 -r now`
   g. Hit the *ESCAPE* key to leave the insert mode in vi
2. While the inittab file is open, use the arrow keys to move up and add a line to require a password for single user mode. By default, RedHat does not require a password for single user mode, which is by default root-level access.

a. Add after "si" option the following, using the *INSERT* key to begin to insert text:
   `--:S:wait:/sbin/sulogin`
b. Hit the *ESCAPE* key to leave the insert mode.
c. Hit the colon (`:`) key, to get into the vi command line at the bottom of the screen.
d. Hit the w key, then the exclamation point (!) and *ENTER* to save the changes.. Then, use the `:q!` (colon-q-exclamation) combination to save the changes to the inittab file. (*ENTER*)
e. Change the file permissions back to the original setting with `chmod 644 inittab`
f. To test this, issue a `shutdown now` (*ENTER*) to reboot the system. When the system comes back up, it will stop during the initialization with the following
   Give root password for maintenance
   (or type Control-D for normal startup):
g. Hit *CTRL-D* to have a normal startup.

Sticky Bits

When this bit is set on a directory, it means that users may not delete or rename other users' files in this directory. This is typically useful for the /tmp directory. Normally, /tmp is world-writeable, enabling any user to delete another user's files. By setting the sticky bit on /tmp, users may only delete their own files from this directory. To set the sticky bit on the /tmp directory, use the command `chmod o+t /tmp` This will be the first example of why ten separate drive devices were created during the installation.

SERVICES changes

First, change this etc/service file permissions from 644 to 600, so that only root can read this file. This is done with the command `chmod 600 /etc/services` For our machine, we are going to edit this file and disable some services. By adding a pound-sign (#) in the first character, the service being described on the line will be disabled. As a start, we disable the echo[i] (7), systat[ii] (11), daytime[iii] (13), qotd (17), chargen[iv] (19), time (37), bootps (67), bootpc (68), gopher (70), pop2 (109), pop3 (110), sunrpc (111), nntp[v] (119), netbios-ns (137), netbios-dgm (138), netbios-ssn (139), talk[vi] (517), nfs[vii] (2049), cvspserver[viii] (2401), netstat[ix] (15), linuxconf[x] (98), and rsh[xi].

Note that after changing the inetd.conf, you need to stop and start the process. This is done by first using the command `ps ax | grep inetd | grep –v grep`
Note the process number at the beginning of the line, such as
577 ? S 0:00 inted –stayalive –reuse –pidfile /var/run/xinetd.pid
To stop/start the inetd process, use the command `kill -HUP 577` (Use the correct process identification number, not necessarily the 577 in this example).

It would be easier to use tcp_wrappers (see later section), instead of commenting out hundreds of unnecessary services in the /etc/services file.

After the security enhancements applied to services, we could connect the server to the network. While all of the security aspects are not covered yet, we need to get some software and also the current patches.

Installation of WU-FTPD

For some reason, the ftp program was not loaded on the system. Using either the RedHat installation CDROM, or going to the directory /pub/redhat/redhat-7.1-en/os/i386/RedHat/RPMS/ RedHat web site, `get wu-ftpd-2.6.1-16.i386.rpm` and `anonftp-4.0-4.i386.rpm`. We downloaded the files from RedHat because the CDROM had already been removed from the system. The logical download directory was used by `cd /usr/src/redhat/RPMS/i386`
After downloading, the installation was done through the RedHat Package Manager program with the command `rpm -ivh wu-ftpd* anonftp*` Changing directory to `/etc/xinetd.d`, open the `wu-ftpd` with vim editor. Change the line `disable=yes` to `disable=no`.

The tcp_wrapper Process

The tcp_wrapper is already installed in the system. For security, we will limit the access of the supported services (chargen, daytime, echo, finger, ntalk, rexec, rlogin, rsh, rsync, talk, telnet, time, and wu-ftpd) to the local network (192.168.100.xxx). The `/etc/hosts.allow` file should have two lines added. The first line should read `ALL:127.` (include a period at the end!) and the next line, added, should read `ALL:192.168.100.` (remember the period). There are trailing periods after each line with the first letting the local use of the processes, while the second line limits external to any IP address from 192.168.150.100.1 to 192.168.100.254[xii]. The etc/hosts.deny file should have a line at the end, `ALL:ALL` These two files will combine to stop all but the local network and local server from using tcp-wrapped services.

Banner changes

One of the most lucrative points of attack for a malicious "cracker" is the information about the system. Quite a bit of information is provided by banners and prompts provided by the operating system. All console logins to the system, whether local console or via telnet, by default, the screen comes up with

```
Red Hat Linux release 7.1 (Seawolf)
Kernel 2.4.3-12 on an i686

GRIMRH login:
```

To change this, edit the etc/rc.d/rc.local file. Comment out the following lines with the # sign:

```
#echo "$R" >> /etc/issue
#echo "Kernel $(uname -r) on $a $SMP$(uname -m)" >> /etc/issue
```

We will add, after these comment-out lines, the following:

```
echo "FOR AUTHORIZED USE ONLY" >> /etc/issue
```

```
echo "Violators may be prosecuted" >> /etc/issue
```

This script, rc.local. recreates two files when the system initializes. The local login screen is stored at `/etc/issue`, while the ftp/telnet/ssh login screen is stored at `/etc/issue.net`.

The /etc/ftpaccess is the configuration file for the wu-ftp program. Open the file `ftpaccess` in the vim editor, add the following lines to the end of the file:

```
# To identify the message given the user before they login,
# you can specify greeting full.  However, we want minimal
# information so specify greeting terse
greeting terse
```

Now, when accessing the server through FTP, we see:

```
Connected to 192.168.100.251.
220 FTP server ready.
User (192.168.100.251:(none)):
```

Instead of the full message of:

```
Connected to 192.168.100.251.
220 GRIMRH.GRIM.COM FTP server (Version wu-2.6.1-16) ready.
User (192.168.100.251:(none)):
```

The file, etc/motd, provides the message after login via local console or telnet. This will be an appropriate place to put a warning about the use of the system. Edit the /etc/motd, adding the following lines:

```
**************************************************************
* USE OF THIS SYSTEM IS RESTRICTED TO AUTHORIZED USERS ONLY. *
*      UNAUTHORIZED ACCESS OR USE IS PROHIBITED BY LAW.      *
*     USE WILL BE MONITORED. VIOLATORS MAY BE PROSECUTED.    *
**************************************************************
```

This banner will appear on telnet and console login screens after a successful password.

SETUID/SETGID File Check

As part of a routine check, identify all files that have the setuid or setgid bits set. Unauthorized use of these settings can provide a malicious user with greater permissions than normal. Our system, as configured, does not have any files with setuid or setgid bits set. This is verified by using the command: `find / -type f -a \( -perm -4000 -o -perm -2000 \) -print`

World-writable Directories

These directories would allow a malicious user to change, modify or add files to the system. Care should be taken to examine the file listing provided by the command `find / -perm -2 -print`.

Password Security

The default password length is five characters. At a minimum, change the password length to eight characters. This is done by editing the `/etc/login.defs` Change the `PASS_MIN_LEN` from 5 to 8, To limit brute attempts to login to the system, we will set the minimum time to wait after a failed login by adding the line `FAIL_DELAY 10` to `the /etc/login.defs`. Also, we will log all unknown failed logins with a line added `LOG_UNKFAIL_ENAB yes`.

Timeout for Inactivity

An open terminal, especially if logged in as root, is an invitation to any user. To set an inactivity limit, edit the `/etc/profile` file. After the HISTSIZE line, add the following line: `TMOUT=1800` This represents 30 minutes (1800 seconds).

Limit the history of shell commands

There is a file for each user, in their home directory, called `.bash_history`. This file continues to grow. By default, this is 1000 lines. To change this, edit the /etc/profile file and change the `HISTSIZE=1000` to `HISTSIZE=20`. Also, we add a limitation on the size of the history with a new line, after the `HISTSIZE` line, adding `HISTFILESIZE=20`. Save with the `:w!` and `:q!` commands.

Limiting root access

In order to protect the system, root should not be able to login remotely. This would be through a remote terminal with rlogin or other programs. Also, ftp and telnet by root would also be restricted. Of primary concern is the passing of the unencrypted root password using these three routes.
To limit the login by root to one terminal, edit the `/etc/securetty` file, and put a number (#) sign in front of all but the `tty1` line. WU-FTP already denies all users with a user/group id less than 99 to access the server with ftp. This denies ftp access to all system users including root.

Setting the root UMASK

The umask command can be used to determine the default file creation mode on your system. For our system, we chose 027 as our umask. This shows us that new executables that are created are given mode 750, which means that the owner can read, write, and execute the binary, while members of the group to which the binary belongs can read and execute it, and all others, cannot read, write, or execute it. This is changed by using vi to change the `/etc/profile` line from `umask 022` to `umask 027`.

Shutdown Unwanted Services

Because of the variety of packages installed on our system, there are a lot of unsecure services running after the installation. Before bringing the system up online, we will disable some services.

The `ntsysv` is a text based utility to show both existing automatic services. In our installation, we unselected the following services

a. apmd (used for monitoring battery status and logging it via syslog)
b. netfs (mounts and unmounts all Network File System (NFS) Samba/SMB (Lan Manager/Windows) and NCP (Netware Core Protocol) mount points)
c. nfslock (protocol for file sharing across TCP/IP networks. Provides NFS file locking functionality)

d. portmap (Manages RPC connections, which are used by protocols such as NFS and NIS. The portmap service must be running on machines which act as servers for protocols which make use of RPC mechanism)
e. sendmail (The mail transport agent which moves mail from one machine to another)
f. sshd (OpenSSH server daemon)

Some of these services will be restarted when they are hardened later in the exercise. Some will be left for security in the future, but will be disabled until secure.

Limit SuperUser capability

At the installation, we created two users (ldgrim and secldg). We want only the user ldgrim can elevate to SuperUser (with the su command). To do this, we edit the `/etc/pam.d/su` file. Modify the first line to read:

```
auth sufficient   /lib/security/pam_rootok.so debug
```

Uncomment (remove the number # character at the beginning of the line) for

```
auth  required          /lib/security/pam_wheel.so use_uid
```

Remember to save the changes with the `:w!` (colon w exclaimation), then exit vim with the `:q!` (colon q exclaimation). The wheel group has a Group ID (GUID) of 10. To verify this, we examine the /etc/group file (possibly with the command `cat /etc/group`). To add the user ldgrim to the wheel group, use the console command `usermod -G10 ldgrim`. This adds the user ldgrim to the group 10 (wheel). We can verify this change, by `cat /etc/group` after the user modification.

OpenSSH security

The Open Secure Shell version 2.5.2p2 is part of the RedHat 7.1 installation. It replaces telnet, ftp, rlogin, rsh, and rcp. However, it first needs to be turned on with the utility ntsysv, used earlier. Just toggle or mark the sshd process as an opened ntsysv service. This will start the most basic, yet secure shell with a ssh-client (such as included with OpenSSH secure shell protocol client (in RedHat 71) or PuTTY (for Windows clients, see www.chiark.greenend.org.uk/~sgtatham/putty/).

There is a configuration file for sshd, which is read both at initialization and termination of an SSH session. It is at `/etc/ssh/sshd_config` For our security purposes, we have added the following line:

```
AllowGroups wheel
```

and changed the X11 line from yes to no, as such:

```
X11Forwarding no
```

As noted in Ulrich Flegel's paper listed in the bibliography, there is a flaw with X11 forwarding over SSH that allows for rogue or malicious capture of an X11-SSH session. There are many more configuration settings in the manual page for sshd, but these will be the initial setup. After making changes to the

configuration, you can stop the sshd service with `/etc/service sshd stop`, then restart with `/etc/service sshd start`.
We will not change the banner that OpenSSH provides a client. The reason, noted in an email[xiii]:

> "i don't see a reason why openssh should say: SSH-1.5-ssh-1.2.27 or SSH-1.5-OpenSSH-1.2.3 if you still run 1.2.2. Changing the version number does not fix bugs. Note that the SSH-1.5 prefix is obligatory. Additionally, clients may use the vendor suffix for bug/feature-compatibility, so it's a bad idea to change this."

Installation of Additional Packages for Web Server

Because we wanted to limit the number of files added to the system, we did not select the setup as a web server. Because of this, we have to install the following packages:

apache-devel-1.3.19-5.i386.rpm	apache-manual-1.3.19-5.i386.rpm
perl-Perl-RPM-0.291-2.i386.rpm	php-4.0.4pl1-9.i386.rpm
php-devel-4.0.4pl1-9.i386.rpm	php-imap-4.0.4pl1-9.i386.rpm
php-ldap-4.0.4pl1-9.i386.rpm	php-manual-4.0.4pl1-9.i386.rpm

Figure 7: WebServer Packages

To get these packages, either remount the cdrom or ftp the files from ftp.redhat.com. For ease, we have kept all additional RPMs (RedHat Package Manager) files in the /usr/src/RedHat/RPMS/i386 directory. With the rpm packages in that directory, at the prompt, type `rpm -Fvh *.i386.rpm` to install all of the packages.

Generating an SSL Key

The OpenSSL is already installed in the setup, but we will generate a random key for the server by typing the following:

```
cd /etc/httpd/conf
make server.key
```

The program will ask for a passphrase or password, which is used to generate the random key. After generating the key, type `mv server.key ssl.key` and overwrite the sample key provided with the installation. To generate a self-certificate, again in the /etc/httpd/conf directory, type `make testcert` This will prompt for a **`PEM pass phrase`**:, then a series of questions:

`Country name (2 letter code) [AU]:` US
`State or Province Name (full name) [Some-state]:` Maryland
`Locality Name (eg. City) []:`Anrold
`Organization Name (eg. Company) [Internet Widgits Pty Ltd]:` Grim Household
`Organizational Unit Name (eg. section) []:` *ENTER* (leave blank)
`Common Name (eg. your name or your server's hostname) []:` GRIMRH.GRIM.COM
`Email address:` ldgrim@grim.com

This will generate the `server.key` in the `/etc/httpd/conf/ssl.key/` directory. The next step, `make testcert` prompts for the same information and creates the self-certifying `server.crt` file for use with Apache SSL operations.

Using Apache Configuration Tool

Before running the tool, issue the following commands to stop any web server daemons:

```
/etc/rc.d/init.d/httpd stop
```

Please note that if you are going to use the Apache Configuration Tool, you must not edit httpd.conf by hand.

Apache WebServer Configuration

It is easier to use the KDE desktop, and the File Manager (Super User Mode) to move the files. Login at the console as `root`, then run `startx`. This will open up the X-Windows KDE interface. The file manager is under the group System. Navigation is similar, both on the desktop and within the File Manager, to the methods used in Windows 9.X/NT programs. Using the Apache Configuration, fill in the server hostname (`GRIMRH.GRIM.COM`) and accept all other defaults by exiting the utility. Exit the KDE manager also, bringing you back to the text console prompt.

Starting the Apache Web Server

Use the `ntsysrv` utility to turn on the web server httpd. Reboot the server, with `shutdown -r now`. After a reboot, the webserver will be online at port 80.

Updating the System

RedHat makes it quite easy to update the system. Connect the server to the internet. Prior to opening the X-windows KDE desktop, run `rhn_register`. The online questions will establish your account. Open up the KDE environment with `startx`. Start the Update Agent, under Services, to being the update process. Using the Update Center, in the KDE environment under system, it was painless to connect to the RedHat network, inventory the system, download and apply the patches. Recommend a reboot to have any changes implimented.

Conclusion

The installation and security steps for a Red Hat 7.1 Linux installation took about three hours from start to finish. At the end of the installation, there was a secure web server and hardened operating system.

While Red Hat has moved to a more secure, default installation; the steps above show some of the more critical steps still need to be performed by hand by the system administrator.

Bibliography

Curry, David, " Improving the Security of your Unix System", 1990, URL:http://www.alw.nih.gov/Security/Docs/unix-security.html

Deraison, Renaud, "Nessus Scanner Plug-in Notes" , 1999, URL: http://cgi.nessus.org/plugins/.

Fenzi, Kevin and Dave Wreski, "Linux Security HOW-TO" v1.1.1, dated 17 March 2000, URL: http://www.europe.redhat.com/documentation/HOWTO/Security-HOWTO-8.php3

Flegel, Ulrich, "The interaction of SSH and X11", September 1997, URL: http://rootshell.com/docs/ssh-x11.ps.gz

Husman, Hans, " Introduction to Denial of Service", 1997, URL:http://www.securityfocus.com/data/library/dos101.txt

Internet Security Systems, "Exploit of x-windows called XF:xcheck-keystroke", 1997, URL: http://xforce.iss.net/static/155.php

Mourani, Gerhard, "_Securing and Optimizing Linux: RedHat Edition", June 7, 2000, URL: http://www.linuxsecurity.com/docs/Securing-Optimizing-Linux-RH-Edition-1_3.pdf

Perkel, Marc, " VIM (1) Vi IMproved, a programmers text editor manual page", URL: http://linux.ctyme.com/man/man2661.htm

SecuriTeam, "Linuxconf contains remotely exploitable buffer overflow", 1999, URL: http://www.securiteam.com/exploits/Linuxconf_contains_remotely_ exploitable_buffer_overflow.html

SecuriTeam, "Who guards your front doors? (A practical guide to securing POP3 under Linux)", 2000, URL: http://www.securiteam.com/unixfocus/5JP01200KY .html

Swan, Jay, "Configuring and using tcpwrappers", undated, URL: http://www.cats.wright.edu/catsweb/ns/osxs_sec/tcpw_install.html

Endnotes

[i] The reason for disabling echo is noted in Nessus comments by “This port is not of any use nowadays, and may be a source of problems, since it can be used along with other ports to perform a denial of service. You should really disable this service.”

[ii] The "systat" service provides useful informations to crackers, such as which processes are running, who is running them, and so on...(from Nessus plugin notes.)

[iii] The reason for disabling the daytime is noted in Nessus testing as “The date format issued by this service may sometimes help an attacker to guess the operating system type. In addition to that, when the UDP version of daytime is running, an attacker may link it to the echo port using spoofing, thus creating a possible denial of service.” This was noted in Bugtraq ID 2395, when the “version of inetd shipped with RedHat Linux 6.2 improperly handles the closing of sockets created in handling requests to inetd's daytime service. As a result, new connections to daytime cause affected versions of inetd to leak resources over time.”

[iv] Nessus comments note “The 'chargen' service should only be enabled when testing the machine. When contacted, chargen responds with some random (something like all the characters in the alphabet in row). When contacted via UDP, it will respond with a single UDP packet. When contacted via TCP, it will continue spewing characters until the client closes the connection. An easy attack is 'pingpong' which IP spoofs a packet between two machines running chargen. They will commence spewing characters at each other, slowing the machines down and saturating the network.”

[v] As late as July 2001, there have been possiblie exploitations of the USENET/Network News Transport Protocol (NNTP).

[vi] talkd is the server that notifies a user that someone else wants to initiate a conversation) Malicious hackers may use it to abuse legitimate users by conversing with them with a false identity (social engineering). In addition to this, crackers may use this service to execute arbitrary code on your system. (Nessus)

[vii] Older versions of nfsd can be used to gain root remotely

[viii] A CVS (Concurrent Versions System) server is installed, and it is configured to have its own password file, or use that of the system. This service it starts as a daemon, listening on port TCP:2401. Knowing that a CVS server is present on the system, gives attackers additional information about the system, such as that this is a UNIX based system, and maybe a starting point for further attacks. (Nessus)

[ix] “The "netstat" service provides useful informations to crackers, since it gives away the state of the active connections.” (Nessus)

[x] According to SecuriTeam, “There is a buffer overflow vulnerability in the Linuxconf package that is shipped with several Linux distributions. The vulnerability may be in the program's handling of HTTP header” (see bibliography)

[xi] From RedHat “Additionally, you really want to disable the rsh/rlogin/rcp utilities, including login (used by rlogin), shell (used by rcp), and exec (used by rsh) from being started in /etc/inetd.conf. These protocols are extremely insecure and have been the cause of exploits in the past.”

[xii] There is an excellent subnet calculator at http://www.hswnetworks.com/ipsubnetcalc.html, which is a United Kingdom based networking consulting firm, HWNetwork.

[xiii] From Markus Friedl, available at http://marc.theaimsgroup.com/?l=openssh-unix-dev&m=95293299506340&w=2

Last Updated: January 23rd, 2019

Upcoming Training

SANS 2019	**Orlando, FL**	**Apr 01, 2019 - Apr 08, 2019**	**Live Event**
SANSFIRE 2019	**Washington, DC**	**Jun 15, 2019 - Jun 22, 2019**	**Live Event**
Community SANS New York SEC506	**New York, NY**	**Jul 15, 2019 - Jul 20, 2019**	**Community SANS**
SANS Network Security 2019	**Las Vegas, NV**	**Sep 09, 2019 - Sep 16, 2019**	**Live Event**
SANS OnDemand	**Online**	**Anytime**	**Self Paced**
SANS SelfStudy	**Books & MP3s Only**	**Anytime**	**Self Paced**