Global Information Assurance Certification Paper

Securing Red Hat Linux 9 as an Apache Web Server, VSFTP Server and MySQL server

GIAC Certified UNIX Security Administrator (GCUX)

Ricky Wald.BA.OXON

December 5, 2003

Section 1 – Table of Contents.

Section 2 - Background & System

Abstract

In this document we will install and then harden a Linux server running:

- Apache Web Server
- FTP (or more accurately a VSFTP) Server
- MySQL Server

On occasions throughout this document, there will be business requirements which force configuration options etc that are against security best practices. However, whilst this can be pointed out to management or whomever the requirement comes from, it does not guarantee that the requirement will be removed or altered. In this document we shall secure the system as best we can given these constraints. Sometimes this is done for demonstration purposes.

This document was written with the concept of focusing on our environment but also being useful to apply the information to other situations, hence the method to obtain the instructions for an action is also included, in addition to the instructions themselves. Furthermore this document follows general hardening methodologies but also the CIS methodologies, with the key difference being that the methodologies are applied to this specific situation. This document avoids using script type command sequences and prefers basic command sequences so that it is clear what is being done, since the focus of this document is on hardening/security rather than learning to script.

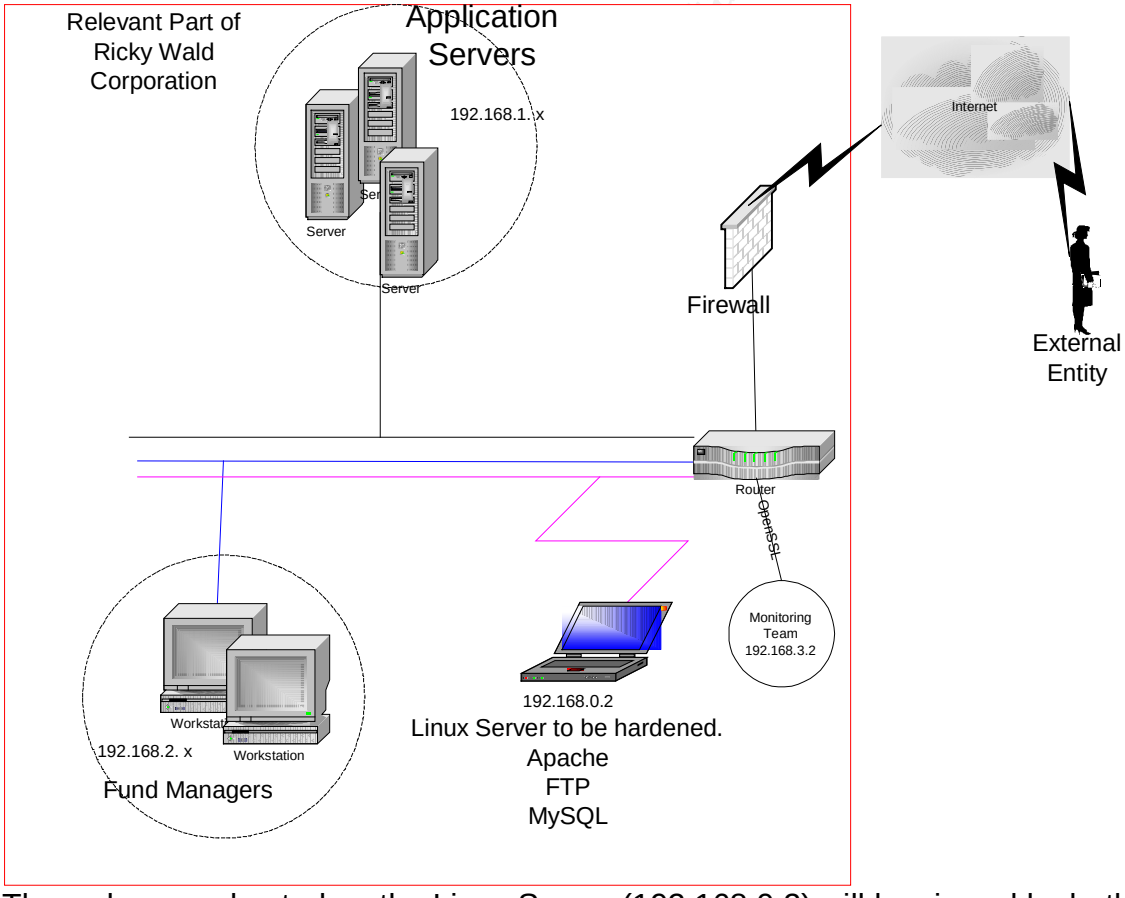
As per the GIAC request, this document is designed to be used by someone with little knowledge of Linux. Furthermore this document follows the attitude that if the document does not repeat security steps, then perhaps the user will be too "lazy" to repeat certain steps, in particular this includes validating md5 hash values and other common bad practices. This is even more important when considering that the user may only look at a certain section of this document or that different users may perform different parts of the installation/hardening.

We will walk through an introduction as an example to highlight where the need for this hardening might arise in a real situation. We will install the Linux operating system. Following on from this, we shall harden the system and use CIS scan, Nessus, Snort and John the Ripper as security tools. We will finish with a list of actions to be performed as part of ongoing maintenance.

Introduction

Ricky Wald Corporation, a strategic consultancy and fund management company, with an annual revenue of £4.75m had to delay their IPO (Initial Public Offering) due to negative publicity involving a security breach in which mathematical algorithms used to value companies was altered and highly sensitive customer data was stolen. The impact of the attack is estimated to cost approximately £1.2m. It has been suggested that the attack was carried out by Derick Huhges, the recently ejected and disgruntled director of sales. Unfortunately the logging set up on the compromised systems was not sufficient and forensic analysis has not produced any leads. In order to restore confidence to both customers and analysts, Ricky Wald Corporation will be implementing improved security whilst demonstrating that the company can cut costs by using open source software wherever possible. You are in charge of installing and hardening a Linux server hosting an Apache Web Server, an FTP Server and MySQL server.

Below is a high level diagram of the relevant part of the network



The web pages hosted on the Linux Server (192.168.0.2) will be viewed by both Fund Managers and a password (application level) controlled subset of external Internet Users. Application servers (192.168.1.x) will use the system (i.e. 192.169.0.2) as an FTP server. The monitoring and support team (i.e. 192.168.3.2) will use OpenSSL to connect to this Linux Server.

Note that ideally we should not have all these services running on a single box, however this is an assumption on which this document is written.

Description of the system

COMPAQ Evo N600c Laptop
Intel(R) Pentium(R) III
Mobile CPU 1200Mhz
1.20GHz
512MB of RAM
3 Button Mouse (PS/2)
1 x Ethernet Network Interface Card

We will be installing:

Item	Location
Red Hat Linux 9 including the packages listed in Package Group Selection screen.	www.redhat.com
openssl-0.9.7c.tar.gz	ftp.openssl.org
httpd-2.0.48.tar.gz	http://httpd.apache.org/download.cgi.
libpcap-0.8.1-316.tar.gz	//www.tcpdump.org/
snort-2.0.2.tar.gz	http://www.snort.org/dl/
Glib-1.2.10.tar.gz	ftp.gimp.org
Gtk+-1.2.10.tar.gz	ftp.gimp.org
Gcc-3.2.2-5.i386.rpm	Red Hat Linux CD2 of 3
sharutils-4.2.1-14.i386.rpm	Red Hat Linux CD3 of 3
cis-linux.tar.gz	www.cisecurity.org
nessus-2.0.9/nessus-installer	www.nessus.org
john-1.6.tar.gz	http://www.openwall.com/john
Patches	As described in Retrieving and installing packages with up2date

Risk Analysis

- External attack from external entities who will need to bypass the checkpoint border firewall.
- Internal attacks from disgruntled employees.
- Physical attack, such as using a CD to boot up into an alternative operating system or a denial of service physical attack such as destroying the power system.

- Attempting to attack running services even if these services are not used by the system on a day-to-day basis.
- Trying to attack by using a weakness in httpd, vsftpd, openssl or mysqld.
- Viruses
- Worms
- Buffer overflow
- Denial Of Service
- Accessing files that the user does not require access to.
- Attacks based on using the default shell of system accounts.
- Attackers outside the application server subnet attempting to attack via vsftpd.
- Man in the middle attacks on the session between the monitoring staff and the Linux server.
- Attacks using known weaknesses for which patches are already released.
- Vulnerabilities that can be detected from scanning tools.
- Attacks based on users mounting devices that the user does not need to.
- Remote attacks automatically giving root privileges.
- Password cracking attacks / Brute force.

We are also going to mitigate the attacks by taking backups, and increase the forensics by taking logging information.

Issues with the Apache version contained by default in Red Hat Linux 9.

If we had simply installed the version of Apache that is shipped with Red Hat Linux 9, then Nessus would have found the following security issues:

Red Hat 9 Apache issue 1 found by Nessus: The remote host appears to be running a version of Apache 2.x which is older than 2.0.48. This version is vulnerable to a bug which may allow a rogue CGI to disable the httpd service by issuing over 4K of data to stderr. To exploit this flaw, an attacker would need the ability to upload a rogue CGI script to this server and to have it executed by the Apache daemon (httpd).

Solution: Upgrade to version 2.0.48 when it is available
See also: http://nagoya.apache.org/bugzilla/show_bug.cgi?id=22030
Risk factor: Low

Red Hat 9 Apache issue 2 found by Nessus (if SSL support was included): Warning found on port https (443/tcp). The remote host appears to be running a version of Apache 2.x which is older than 2.0.43. This version allows an attacker to view the source code of CGI scripts via a POST request made to a directory

with both WebDAV and CGI enabled. Note that Nessus solely relied on the version number of the remote server to issue this warning. This might be a false positive

Solution: Upgrade to version 2.0.43
See also: http://www.apache.org/dist/httpd/CHANGES_2.0
Risk factor: Medium
CVE: CAN-2002-1156, CAN-2003-0083
BID: 6065

Red Hat 9 Apache issue 3 found by Nessus (if SSL support was included):
Warning found on port https (443/tcp). The remote host appears to be running a version of Apache 2.x which is older than 2.0.45. This version is vulnerable to various flaws:

- There is a denial of service attack which may allow
an attacker to disable this server remotely

- The httpd process leaks file descriptors to child processes,
such as CGI scripts. An attacker who has the ability to execute
arbitrary CGI scripts on this server (including PHP code) would
be able to write arbitrary data in the file pointed to (in particular,
the log files)

Solution: Upgrade to version 2.0.45
See also: http://www.apache.org/dist/httpd/CHANGES_2.0
Risk factor: Medium
CVE: CAN-2003-0132
BID: 7254, 7255

Red Hat 9 Apache issue 4 found by Nessus (if SSL support was included):
Warning found on port https (443/tcp). The remote host appears to be running a version of Apache 2.x which is older than 2.0.4. This version is vulnerable to various flaws :

- There is a denial of service vulnerability which may allow
an attacker to disable basic authentication on this host

- There is a denial of service vulnerability in the mod_dav module
which may allow an attacker to crash this service remotely

Solution: Upgrade to version 2.0.46
See also: http://www.apache.org/dist/httpd/CHANGES_2.0
Risk factor: Medium
CVE: CAN-2003-0245, CAN-2003-0189
BID: 7723, 7725

Red Hat 9 Apache issue 5 found by Nessus (if SSL support was included): Warning found on port https (443/tcp). The remote host appears to be running a version of Apache 2.x which is older than 2.0.47. This version is vulnerable to various flaws which may allow an attacker to disable this service remotely and/or locally.

Solution: Upgrade to version 2.0.47
See also: http://www.apache.org/dist/httpd/CHANGES_2.0
Risk factor: Medium
CVE: CAN-2003-0192, CAN-2003-0253, CAN-2003-0254
BID: 8134, 8135, 8137, 8138

Physical location of the Linux system

To mitigate the risk of a physical attack the Linux box will be located in a secure datacentre with the following security/availability measures in place.

- Monitored air handling and environment (Air replacement, Temperature, Humidity, Pressure).
- Internal motion activated cameras.
- 24 Hour security guards.
- Remote site monitoring.
- Bomb resistant walls/ceilings with thick metal plates.
- All building access systems have full audit trails.
- External surveillance cameras.
- Infrared beam surrounding the building.
- Infrared cameras.
- Vibration sensors on all glass.
- Monitored fire detectors (both optical and ionization).
- Monitored fire extinguishers and standalone fire extinguishers (Halon gas).
- Direct link to local fire services.
- Perimeter corridor surrounding main datacentre area.
- Smart card access to secure rooms.
- PIN access to secure rooms.
- Security barriers to parking areas.
- Large site UPS modules.
- Large site power generators.
- Fuel tank.
- Robotic tape systems.
- Accompanied access to secure rooms.
- Faraday cage (to prevent electromagnetic waves escaping).
- Floor to ceiling swivel doors.

Section 3 - Step by step guide to installation of the Linux operating system.

Start with trusted copies of the Linux installation CDs. For example assuming you trust Red Hat, CDs can be obtained directly from Red Hat. This is to prevent an installation being performed with a maliciously or otherwise altered version of Linux. For example if you download Linux over the Internet, there is a possibility that a man in the middle attack can be carried out such that a version of Linux is installed with purposely-included back doors. The MD5 hash value can also be checked.

1. Insure that CD-ROM is the primary boot device. Place installation CD1 into the CD bay of the Linux Server (or rather the soon to be Linux Server) and restart the machine.

2. You will be presented with a Red Hat Linux 9 screen prompting
To install or upgrade Red Hat Linux in graphical mode, press the <ENTER> key.
To install or upgrade Red Hat Linux in test mode, type linux text <ENTER>
Use the function keys listed below for more information
<output omitted>
boot: **<ENTER>**

3. The CD Found page is displayed.

CD Found

To begin testing the CD media before installation press OK. Choose Skip to skip the media test and start the installation.

OK

4. The Media Check screen is displayed.

Choose “Test” to test the CD currently in the drive, or “Eject CD” to eject the CD and insert another for testing.

Test

5. The Media Check progress bar is displayed.

Checking “Red Hat Linux 9 disc 1” …-

6. The Media Check Result is displayed.

The Media check of the image: Red Hat Linux 9 disc 1 is complete, and the result is: PASS. It is OK to install from this media.

OK

7. The Media Check page is displayed.

If you would like to test additional media, insert the next CD and press “Test”. You do not have to test all CDs, although it is recommended you do so at least once.

To begin the installation process, insert CD #1 into the driver and press “Continue”.

Insert the next CD (i.e. CD#2)

Test

8. The Media Check progress bar is displayed.

Checking “Red Hat Linux 9 disc 2” …-

9. The Media Check Result is displayed.

The Media check of the image: Red Hat Linux 9 disc 2 is complete, and the result is: PASS. It is OK to install from this media.

OK

10. The Media Check page is displayed.

If you would like to test additional media, insert the next CD and press “Test”. You do not have to test all CDs, although it is recommended you do so at least once.

To begin the installation process, insert CD #1 into the driver and press “Continue”.

Insert the next CD (i.e. CD#3)

Test

11. The Media Check progress bar is displayed.

Checking “Red Hat Linux 9 disc 3” …-

12. The Media Check Result is displayed.

The Media check of the image: Red Hat Linux 9 disc 3 is complete, and the result is: PASS. It is OK to install from this media.

OK

13. The Media Check page is displayed.

If you would like to test additional media, insert the next CD and press “Test”. You do not have to test all CDs, although it is recommended you do so at least once.

To begin the installation process, insert CD #1 into the driver and press “Continue”.

Insert CD#1

Continue

14. Anaconda, (the Red Hat Linux system installer) will start.

15. The “Welcome to Red Hat Linux” graphical installation screen will be displayed. Click **Next**.

16. The “Language Selection” screen is displayed. Use the touch pad and the left touch pad button to highlight the appropriate language. In this case the default selection **English (English)** is highlighted. Click **Next**.

17. The “Keyboard Configuration” screen is displayed. Highlight the correct keyboard for example **United Kingdom**. Click **Next**.

18. The “Mouse Configuration” screen is displayed. Highlight the appropriate Mouse, which for the hardware being used in this guide is **3 Button Mouse (PS/2)**. Click **Next**.

19. Note that if a version of Linux is already installed, you will be prompted to select whether you want an upgrade or a new installation, in which case you should select a new installation. The “Installation Type” screen is displayed. Left click the “**Custom**” radio button. Click **Next**.

20. The “Disk Partitioning Setup” screen is displayed. Select the “Manually partition with Disk Druid” radio button. Click **Next**.

21. The Disk Setup screen is displayed. The device table on the bottom right of the screen contains the following information:

- Device
- Mount Point/RAID/Volume
- Type
- Format
- Size (MB)
- Start
- End

For the devices, the options available to you are:

- New – Used to create a new partition. When New is clicked, the Add Partition dialogue box will be displayed, where the Mount Point, File System Type, Size, Size Options etc are chosen.
- Edit – Used to edit the details of an already created partition
- Delete – Used to delete a partition
- Reset – Used to go back without making the changes
- RAID – (Redundant Array of Inexpensive Disks) is used if you wish to spread data over multiple disks to gain benefits in redundancy or speed of writing to disk. Some examples of RAID are mirroring (RAID 1) or parity bits (RAID 5) for increased redundancy and stripping (RAID 0) for increased speed. http://www.acnc.com/04_01_00.html provides an explanation of the different RAID levels.
- LVM (Logical Volume Manager) as its name suggests allows the management of multiple disks logically rather than physically. For example by combining multiple physical disks, a volume is not limited by the physical size of a single disk.

22. Click **New**.

Click the down arrow for the Mount Point selection. Click **/** (root).

For the File System Type, leave the default selection: **ext3**
For the Allowable Drives, use the default selection where the **hda 28616MB** has the check box ticked.

For the size, increase the value to **4000MB**.

Leave this partition as **Fixed size**.

Click the "**Check for bad blocks**" check box.

Click OK so that we will be creating a partition with the following details.

Device	Mount Point/RAID/Volume	Type	Format	Size (MB)
/dev/hda1	/	Ext3	Yes	4000

22. As we did in step 21, create new partitions as follows:

Device	Mount Point/RAID/Volume	Type	Format	Size (MB)
/dev/hda1	/	Ext3	Yes	4000
/dev/hda8	/var	Ext3	Yes	2000
/dev/hda7	<Not Applicable>	swap	Yes	2000 (notice this more than double the RAM size)
/dev/hda6	/home	Ext3	Yes	2000
/dev/hda5	/usr	Ext3	Yes	2000
/dev/hda2	/usr/local	Ext3	Yes	4000
/dev/hda3	/tmp	Ext3	Yes	2000

23. Once you have created the above partitions, click **Next**.

24. The "Boot Loader Configuration" screen is displayed.

Leave "**GRUB boot loader**" (default) as the boot loader.

In our example, there is only one operating system installed, so leave **Red Hat Linux** as the default operating system to boot from.

Click the "**Use a boot loader password**" checkbox. The "Enter Boot Loader Password" dialogue box will appear. Type your password into the Password and Confirm fields. Choose a strong password, for example a mixture of upper case, numeric, alphabetic characters greater than 10 characters etc. Linux states that "a boot loader password prevents users from changing options passed to the kernel." Click **OK** to close the dialogue box.

Leave the "Configure advanced boot loader options" unchecked.

Click **Next**.

25. The "Network Configuration" screen appears.

If you will be statically allocating the IP address. For the Ethernet connection (called eth0), click **Edit** so that the "**Edit Interface eth0**" dialogue box appears, allowing you to enter the IP Address and Netmask by un-checking "**Configure using DHCP**". Leave "**Active on Boot**" checked.

If you are not using DHCP, in the Miscellaneous section, you can enter the Gateway, Primary DNS, Secondary DNS and Tertiary DNS. If DHCP were going to be used, you would leave the "**Configure using DHCP**" check box ticked. Leave "**Active on Boot**" checked.

For the Set the hostname section, click the "**manually**" radio button and type the hostname, for example "**GIAC-Linux**". Note that GIAC-Linux is being used as an illustration but in reality you would not use this hostname as it gives away too much information, being that the operating system is Linux.

Click **Next**.

26. The Firewall Configuration screen is displayed.

Select the No Firewall radio button, rather than High or medium.

Click **Next**.

27. The Additional Language Support screen is displayed.

Click the **English (Great Britain)** radio button.

Please choose the appropriate settings for your situation. In our example we do not need additional language support, so uncheck English (USA) (which is selected by default).

Click **Next**.

28. The Time Zone Selection screen is displayed.

In our example, move your curser over **London** and click. You will see that Europe/London is selected in the lower pane of the screen.

Click **Next**.

29. The Set Root Password screen is displayed.

Enter your root password in both the Root Password and Confirm input boxes. Please choose a strong password as in the boot loader password example.

Click **Next**.

30. The Authentication Configuration screen is displayed.

Leave the default options, which are as follows:

- Enable MD5 passwords checked
- Enable shadow passwords checked. Note that old versions of Unix used to store passwords in clear text in the /etc/password file. Thus anyone with read access to this file had control over the system. Recent versions of Unix/Linux have the password field in /etc/password replaced with an "x" and use /etc/shadow to store passwords in an encrypted form. If someone has access to /etc/shadow, this can be cracked by brute force for example with tools such as John the Ripper (and L0phtCrack for NT), although this can take a long time with strong passwords and is far more secure than not using the shadow option.
- Enable NIS is not checked.
- Enable LDAP is not checked.
- Enable Kerberos is not checked.
- Enable NIS Authentication is not checked.

Click **Next**.

Package Group Selection screen

31. The Package Group Selection screen is displayed.

X Window System	Standard packages plus: XFree86-xdm, firstboot, gdm, redhat-config-date, redhat-config-network, redhat-config-packages, redhat-config-printer-gui, redhat-config-services, redhat-config-soundcard, rhn-applet, switchdesk, up2date-gnome, usermode-gtk. Note that even though from a security perspective X Windows should not be installed, this is a business requirement that we have been given.
GNOME Desktop Environment	Both this and KDE are the graphical, i.e. non command line methods to interface with the Linux operating system. If a graphical interface is required (which it is in this example), then select either GNOME or KDE, but not both. For this system we shall use GNOME. This is a business requirement. Checked for the standard packages, gnome-vfs2-extras and eog. Uncheck all other extra packages.
KDE Desktop Environment	Unchecked.
Editors	Checked. Leave vim-enhanced checked but uncheck all other extra packages.
Engineering and Scientific	Unchecked.
Graphical Internet	Checked, but out of the extra packages, only mozilla and mozilla-psm should be checked.

Text-based Internet	Unchecked.
Office/Productivity	Unchecked.
Sound and Video	Unchecked.
Authoring and Publishing	Unchecked.
Graphics	Unchecked.
Games and Entertainment	Unchecked.
Server Configuration Tools	Checked, but out of the extra packages, only redhat-config-printer, redhat-config-printer-gui and redhat-config-securitylevel should be checked.
Web Server	Unchecked.
Mail Server	Unchecked.
Windows File Server	Unchecked.
DNS Name Server	Unchecked.
FTP Server	Checked.
SQL Database Server	Checked. Make sure that the mysql-server extra package is checked.
News Server	Unchecked.
Network Servers	Unchecked.
Development Tools.	Unchecked. Note that we will install gcc by directly using the RPM contained on the Red Hat Linux CDs.
Kernel Development	Unchecked.
X Software Development.	Unchecked.
GNOME Software Development	Unchecked.
KDE Software Development	Unchecked.
Administration Tools	Checked for redhat-config-keyboard, redhat-config-rootpassword, redhat-config-packages, redhat-config-date, redhat-config-language and redhat-config-soundcard only.
System Tools	Unchecked.
Printing Support	Checked. This is a business requirement.

Click **Next**.

32. The About to Install screen is displayed.

Click **Next**.

33. The Installing Packages screen is displayed.

Initially you will see progress bars for Checking for bad blocks, as we selected this option when choosing our partitions.

Progress bars will be displayed.

34. The Boot Diskette Creation screen is displayed.

Two options will be given:

- Yes, I would like to create a boot diskette.
- **No, I do not want to create a boot diskette**.

Because the hardware that we are using does not have a floppy drive (it does not even have a CD writer), click the second radio button. Note that we will be taking backups as described under Backups in section 8.

Click **Next**.

35. The Graphical Interface (x) Configuration screen is displayed.

Highlight the **ATI Radeon Mobility M6**.

For the Video card RAM, in this example, **32MB** should be selected.

Click **Next**.

36. The Monitor Configuration screen is displayed.

Leave the Unprobed Monitor selected.

Leave the Horizontal Sync and Vertical Sync frequencies as their default values, which in this example is 31.5-37.9 and 50-70 Hz respectively.

Click **Next**.

37. The Customize Graphics Configuration screen is displayed.

Leave the following with their default values.

Color Depth	True Color (24 bit)
Screen Resolution	800x600

Select the login type as **Graphical**.

Click **Next**.

38. Next the “Congratulations screen is displayed.”

Click **Exit**.

Red Hat Setup Agent Wizard

39. The Welcome, Red Hat Setup Agent wizard is displayed.

Click **Forward**.

40. The User Account screen is displayed.

Fill in the following fields:

- Username
- Full Name
- Password
- Confirm Password

This user account can be used for activities that do not require administrative privileges.

Click **Forward**.

41. The Date and Time screen is displayed.

If you do not intend to use an NTP server: In the Date section on the left hand pane, click the correct date. On the right hand pane under the Time section, select the correct Hour, Minute and Second. We will use this in our example.

Alternatively you can use an NTP server by clicking the **Enable Network Time Protocol** check box. Next you would need to enter the Server details.

Click **Forward**.

42. The Sound Card screen is displayed.

Click **Play Test Sound**.

Assuming you correctly hear the sound click **Yes** to the dialogue box.

Click **Forward**.

43. The Red Hat Network screen is displayed.

You can choose to either:

- Yes, I want to register my system with Red Hat Network.
- **No, I do not want to register my system**.

Choose the latter option, as we will register GIAC-Linux later.

Click **Forward**.

44. View the “General” Tab

Leave the default “Red Hat Network Server” option selected.

Leave the following check boxes unchecked.

- Enable HTTP Proxy.
- Use Authentication.

NB. In this example a Proxy is not used.

45. The Retrieval/Installation tab is displayed.

Click “**Use CPG to verify package integrity**”.

Leave the default option for the Package Storage Directory, which is **/var/spool/up2date**

46. Next you will be informed that:

“Your CPG keyring does not contain the Red Hat, inc Public key. Without it, you will be unable to verify that packages update agent downloads are securely signed by Red Hat. Your update agent options specify that you want to use CPG. Install key.”

47. Click **Yes**.

48. The Additional CDs screen is displayed.

Click **Forward**.

49. The Finish Setup screen is displayed.

Click **Forward**.

Section 4 – Post Installation

Openssl

Installing Openssl

Use the mkdir command to create a new directory /openssl, and use the chmod command to change the permissions to 700.
Now ftp openssl onto our host machine in this directory.

```
[root@GIAC-Linux openssl]# ftp ftp.openssl.org
Connected to ftp.openssl.org (195.27.176.155).
<output omitted>
230-  Source Location:
230-    ftp://ftp.openssl.org/source/ ..................................... CH
230-  Mirror Locations:
<output omitted>
230-    ftp://ftp.si.uniovi.es/mirror/OpenSSL/ ......................... ES
<output omitted>
Using binary mode to transfer files.
ftp> cd source
<output omitted>
250    it was last modified on Mon Mar 22 18:28:40 1999 - 1680 days ago
ftp> hash
Hash mark printing on (1024 bytes/hash mark).
ftp> get openssl-0.9.7c.tar.gz <output omitted>
##<output omitted>##
ftp> get openssl-0.9.7c.tar.gz.md5
##<output omitted>##
ftp>
```

Next, view the md5 hash that we downloaded

```
[root@GIAC-Linux openssl]# more ./openssl-0.9.7c.tar.gz.md5
md5  : c54fb36218adaaaba01ef733cd88c8ec  openssl-0.9.7c.tar.gz
```

Next compute the md5 hash value of openssl-0.9.7c.tar.gz

```
[root@GIAC-Linux snort]# md5sum openssl-0.9.7c.tar.gz
c54fb36218adaaaba01ef733cd88c8ec  openssl-0.9.7c.tar.gz.md5
```

Notice that the values match.

Next gunzip, then extract the files from the archive using the tar xf command.

[root@GIAC-Linux openssl]# gunzip /openssl/openssl-0.9.7c.tar.gz
[root@GIAC-Linux openssl]# tar xf /openssl/openssl-0.9.7c.tar

Move to the directory /openssl/openssl-0.9.7c.

Next read the installation instructions.
[root@GIAC-Linux openssl-0.9.7c]# more /openssl/openssl-0.9.7c/INSTALL
Thus we follow the following command sequences to perform the installation of openssl.
[root@GIAC-Linux openssl-0.9.7c] ./config
[root@GIAC-Linux openssl-0.9.7c] make
[root@GIAC-Linux openssl-0.9.7c] make test
[root@GIAC-Linux openssl-0.9.7c] make install

Verifying ssh connection using password

ssh as user rickywald from the client machine to the server GIAC-Linux.
[rickywald@client /]$ ssh GIAC-Linux

Because the server GIAC-Linux has not yet authenticated to the client, i.e. that the client trusts server, you as the client are prompted to trust the server. You are presented with the fingerprint/hash of the server's digital certificate.

The authenticity of host ' giac-linux (x.x.x.x)' can't be established.
RSA key fingerprint is 5d:27:3b:1e:22:76:7c:f0:eb:20:db:94:3f:df:1d:74.

Check that the fingerprint does indeed match that on the server. If so, type Yes.

Are you sure you want to continue connecting (yes/no)? Yes
Warning: Permanently added 'giac-linux' (RSA) to the list of known hosts.

rickywald@giac-linux's password:
warning: No xauth data; using fake authentication data for X11 forwarding.
/usr/X11R6/bin/xauth: creating new authority file /home/wald/.Xauthority
[rickywald@GIAC-Linux rickywald]$

We have successfully connected from the client to the server (GIAC-Linux) using ssh.

Exit from the ssh session by typing "exit".

Apache

Installing Apache

Apache 2.0.48 (httpd-2.0.48.tar.gz and the MD5) can be downloaded from http://httpd.apache.org/download.cgi. Place these files in a new folder called /Apache. Obviously any location can be used, but using the same folder locations assists in communication within this document.

Check that the MD5 hash value of httpd-2.0.48.tar.gz from the appache.org website matches the MD5 hash value calculated on our host machine GIAC-Linux.

```
[root@GIAC-Linux root]# cd /Apache
[root@GIAC-Linux Apache]# cat httpd-2.0.48.tar.gz.md5
63f16638c18b140b649fab32b54d7f9c  httpd-2.0.48.tar.gz
[root@GIAC-Linux Apache]# md5sum httpd-2.0.48.tar.gz
63f16638c18b140b649fab32b54d7f9c  httpd-2.0.48.tar.gz
```
Notice that the md5 values do indeed match.

Now we gunzip then extract the httpd files from the zipped and tarred archive.

```
[root@GIAC-Linux Apache]# gunzip httpd-2.0.48.tar.gz
[root@GIAC-Linux Apache]# tar xf httpd-2.0.48.tar
```

The installation instructions from Apache.org can be viewed as follows.
```
[root@GIAC-Linux httpd-2.0.48]# more ./INSTALL
```

We will use the "configure" command without the --prefix tag so that we use the default location which is /usr/local/apache2.
```
[root@GIAC-Linux httpd-2.0.48]#  ./configure
<output omitted>
[root@GIAC-Linux httpd-2.0.48]#  make
<output omitted>
[root@GIAC-Linux httpd-2.0.48]#  make install
<output omitted>
```

Now start the apache server.
```
[root@GIAC-Linux httpd-2.0.48]# /usr/local/apache2/bin/apachectl start
httpd: Could not determine the server's fully qualified domain name, using
127.0.0.1 for ServerName
```

Checking the Apache Web Server is running.

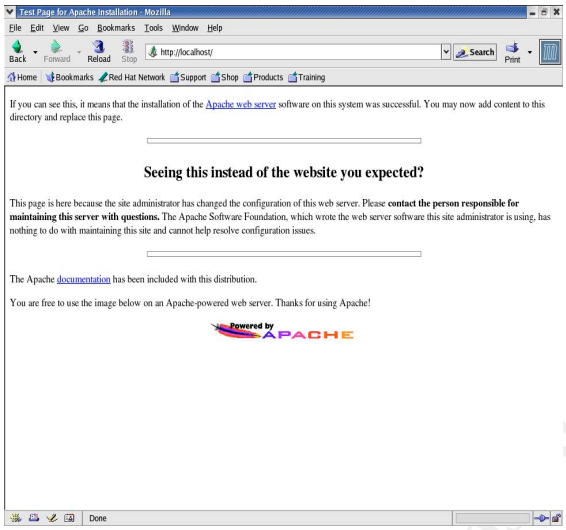
1. If you have not already done so, start the machine.

2. Enter the root username and password to login.

3. Next test that the Apache web server is running by:

From the GUI, click the Mozilla icon on the desktop bar on the bottom left of the screen. In the web browser, type http://localhost/

You should receive a test page confirming that the Apache Web server installed at this site is working properly.



Running Apache at startup

[ASIDE: If you try to use ntsysv to get Apache to start at boot, httpd will not be listed. httpd will not listed upon typing “chkconfig --list” either. Note that: The installation was done without using an rpm.]

Creating symbolic links so that Apache will start on startup:

Ensure that service control script for the run levels, 2, 3, 4 and 5 (located at /etc/rc#.d where # represents the run level) includes the service control script for httpd, which is called S85httpd. This service control script (S85httpd) is really a pointer to the apache control script (apachectl).

```
[root@GIAC-Linux httpd-2.0.48]# cd /etc/rc.d/init.d
[root@GIAC-Linux init.d]# ln -s /usr/local/apache2/bin/apachectl
/etc/rc.d/init.d/httpd
[root@GIAC-Linux init.d]# ln -s /usr/local/apache2/bin/apachectl
/etc/rc.d/rc2.d/S85httpd
```

```
[root@GIAC-Linux init.d]# ln -s /usr/local/apache2/bin/apachectl
/etc/rc.d/rc3.d/S85httpd
[root@GIAC-Linux init.d]# ln -s /usr/local/apache2/bin/apachectl
/etc/rc.d/rc4.d/S85httpd
[root@GIAC-Linux init.d]# ln -s /usr/local/apache2/bin/apachectl
/etc/rc.d/rc5.d/S85httpd
[root@GIAC-Linux init.d]#
```

The “S” stands for start rather than “K” for stop in the above symbolic links.

Check that for run level 2 the Apache pointer in /etc/rc2.d correctly points to the Apache control script (which is apachectl).

```
[root@GIAC-Linux init.d]# ls -n /etc/rc2.d/S85httpd
lrwxrwxrwx    1 0        0              15 Oct 21 15:28 /etc/rc2.d/S85httpd ->
/usr/local/apache2/bin/apachectl
```

This check can be repeated for the other run levels.

You can view the apache control script using the cat command as follows.

```
[root@GIAC-Linux init.d]# cat /etc/rc2.d/S85httpd
<Output omitted>
```

Check that Apache is running at startup by rebooting the machine and following the instructions in Checking the Apache Web Server is running.

MySQL Server

[Note that MySQL was installed using the version already contained in Red Hat Linux 9, however we could have used the instructions found at http://www.mysql.com/documentation/mysql/bychapter/manual_Installing.html#Linux-RPM to download the RPMs found at http://www.mysql.com/downloads/mysql-4.0.html and then validating the checksum. The instructions for checking the MD5 hash can be found at http://www.mysql.com/documentation/mysql/bychapter/manual_Installing.html#Verifying_Package_Integrity but this basically involves using the “shell> md5sum <package>” command.]

The author has translated the instructions to fit this document, but the information was learnt from http://www.mysql.com/documentation/mysql/bychapter/manual_Installing.html#Post-installation

Setting up MySQL server.

First we are going to create the database. The script to accomplish this is located in /usr/bin. This script also creates a test database and it is called mysql_install_db.

```
[root@GIAC-Linux bin]# /usr/bin/mysql_install_db
Preparing db table
Preparing host table
Preparing user table
Preparing func table
Preparing tables_priv table
Preparing columns_priv table
Installing all prepared tables
031022  9:47:48  /usr/libexec/mysqld: Shutdown Complete


To start mysqld at boot time you have to copy support-files/mysql.server
to the right place for your system

PLEASE REMEMBER TO SET A PASSWORD FOR THE MySQL root USER !
This is done with:
/usr/bin/mysqladmin -u root  password 'new-password'
/usr/bin/mysqladmin -u root -h GIAC-Linux  password 'new-password'
See the manual for more instructions.

You can start the MySQL daemon with:
cd /usr ; /usr/bin/safe_mysqld &

You can test the MySQL daemon with the benchmarks in the 'sql-bench'
directory:
cd sql-bench ; run-all-tests

Please report any problems with the /usr/bin/mysqlbug script!

The latest information about MySQL is available on the web at
http://www.mysql.com
Support MySQL by buying support/licenses at https://order.mysql.com

[root@GIAC-Linux bin]#
```

Next we start the MySQL daemon (called mysqld).

```
 cd /var/lib/mysql

[root@GIAC-Linux mysql]# /usr/bin/safe_mysqld &
```

```
[1] 1397
[root@GIAC-Linux mysql]# Starting mysqld daemon with databases from
/var/lib/mysql
```

For easy reference, here is a list of the locations of some important mysqld files.
mysql log file = /var/log/mysqld.log
mysql conf file = /etc/my.cnf

ASIDE: The following command will provide useful information, including the default values.

```
 [root@GIAC-Linux libexec]# /usr/libexec/mysqld --help | more
```

Validation that the MySQL server is running.

Now we are going to see which databases there are. This will be a validation that the MySQL server is running.

```
[root@GIAC-Linux root]# mysqlshow
+-----------+
| Databases |
+-----------+
| mysql     |
| test      |
+-----------+
```

This demonstrates that there are two databases called mysql and test. Now let us see what tables are displayed in the mysql database:

```
[root@GIAC-Linux root]# mysqlshow mysql
Database: mysql
+--------------+
|    Tables    |
+--------------+
| columns_priv |
| db           |
| func         |
| host         |
| tables_priv  |
| user         |
+--------------+
[root@GIAC-Linux root]#
```

Next we will shutdown the mysql daemon.

```
[root@GIAC-Linux root]# mysqladmin -u root shutdown
[root@GIAC-Linux root]#
```

Running MySQL server at startup.

We will ensure that the MySQL server is started automatically on startup.

First we see at which run levels mysqld is running.

```
[root@GIAC-Linux /]# chkconfig --list mysqld
mysqld          0:off   1:off   2:off   3:off   4:off   5:off   6:off
```

You will notice that mysqld is not started automatically at any run level.
Thus we will add run levels 2, 3, 4 and 5 to those at which mysqld will be started.

```
[root@GIAC-Linux /]# chkconfig --level 2345 mysqld on
```

Next we shall check that the modifications have been made.

```
[root@GIAC-Linux /]# chkconfig --list mysqld
mysqld          0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

Notice that mysqld will be started upon entering run levels 2, 3, 4 and 5.

We shall reboot the server as an extra validation.

Type mysqlshow to see which databases are available and hence check that mysqld is running.

```
[root@GIAC-Linux root]# mysqlshow
+-----------+
| Databases |
+-----------+
| mysql     |
| test      |
+-----------+
[root@GIAC-Linux root]#
```

VSFTP Server

[Note that in this example we have used the vsftp server version 1.1.3 contained by default in Red Hat Linux 9. However we could have obtained this open source software from ftp://vsftpd.beasts.org/users/cevans/ and then followed the instructions found at ftp://vsftpd.beasts.org/users/cevans/untar/vsftpd-1.2.0/INSTALL.]

The instructions to set up vsftpd with xinetd.d can be found at ftp://vsftpd.beasts.org/users/cevans/untar/vsftpd-1.2.0/EXAMPLE/INTERNET_SITE/README
With the example xinetd.d configuration file (called vsftpd.xinetd) located at ftp://vsftpd.beasts.org/users/cevans/untar/vsftpd-1.2.0/EXAMPLE/INTERNET_SITE/vsftpd.xinetd.

Setting up the VSFTP Server

Hypothesize that we have come onto this Linux box and we have no knowledge of VSFTP servers. A good step will be to try and find where the ftp files have been installed by Red Hat Linux. Thus, use the "locate" command to determine this.

```
[root@GIAC-Linux /]# locate ftpd
/usr/sbin/vsftpd
/usr/share/doc/vsftpd-1.1.3
/usr/share/doc/vsftpd-1.1.3/Changelog
/usr/share/doc/vsftpd-1.1.3/AUDIT
/usr/share/doc/vsftpd-1.1.3/BUGS
/usr/share/doc/vsftpd-1.1.3/INSTALL
/usr/share/doc/vsftpd-1.1.3/FAQ
/usr/share/doc/vsftpd-1.1.3/README.security
/usr/share/doc/vsftpd-1.1.3/LICENSE
/usr/share/doc/vsftpd-1.1.3/README
....
```

We notice that a lot of the documentation is contained in /usr/share/doc/vsftpd-1.1.3, so we browse to that location.

```
[root@GIAC-Linux /]# cd /usr/share/doc/vsftpd-1.1.3/
[root@GIAC-Linux vsftpd-1.1.3]# ls
AUDIT      FAQ     README          SECURITY  TODO
BUGS       INSTALL  README.security  SIZE      TUNING
Changelog  LICENSE  REWARD          SPEED     vsftpd.xinetd
[root@GIAC-Linux vsftpd-1.1.3]#
```

We will use the INSTALL file as a starting point to learn how to start and configure the FTP server.

```
[root@GIAC-Linux /]# cd /usr/share/doc/vsftpd-1.1.3/
[root@GIAC-Linux vsftpd-1.1.3]# more ./INSTALL
INSTALL
=======
```

This file details how to build and install / run vsftpd from the vsftpd distribution .tar.gz file.

<output omitted>

Next we shall notice that the “nobody” user and the “ftp user” are already created (from the installation of Red Hat Linux), thus we do not need to perform this part of the ./INSTALL instructions.

```
[root@GIAC-Linux root]# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
<output omitted>
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
<output omitted>
apache:x:48:48:Apache:/var/www:/sbin/nologin
mysql:x:27:27:MySQL Server:/var/lib/mysql:/bin/bash
rickywald:x:500:500:Ricky Wald:/home/rickywald:/bin/bash
[root@GIAC-Linux root]#
```

Notice that both the “nobody” and “FTP User” do exist. Also note that the FTP user has a home directory of /var/ftp. FTP User will be used for anonymous FTP access.

At this stage we could for security reasons check the permissions and ownership of the FTP User home directory /var/ftp. However for the purposes of illustration we will make this change after we have taken a “not secure” baseline reading. This is more relevant if we were restricting ftp access to a single user or group. This will become clearer in the sections that follow.

You will notice the Red Hat Linux installed the bin files in /usr/sbin/vsftpd and the example configuration file in /etc/vsftpd/vsftpd.conf.

Copy the example configuration file /etc/vsftpd/vsftpd.conf to the location /etc

```
[root@GIAC-Linux /]# cp  /etc/vsftpd/vsftpd.conf   /etc
```

VSFTPD in standalone mode

As a demonstration, first let us make sure that the configuration file is such that “listen=NO” is displayed.

```
[root@GIAC-Linux /]# vi /etc/vsftpd.conf
# Example config file /etc/vsftpd.conf
#
# The default compiled in settings are very paranoid. This sample file
```

```
# loosens things up a bit, to make the ftp daemon more usable.
<output omitted>
listen=NO
<output omitted>
```

Fail to start vsftpd in standalone mode as follows:

```
[root@GIAC-Linux /]# /usr/sbin/vsftpd &
[1] 1708
[root@GIAC-Linux /]# 500 OOPS: vsftpd: does not run standalone, must be
started from inetd
q
bash: q: command not found
[1]+  Exit 1                  /usr/sbin/vsftpd
[root@GIAC-Linux /]#
```

Notice that we failed to run vsftpd in standalone mode.

We will need to re-modify the vsftpd configuration file to allow vsftpd to run in standalone mode.

Let us modify the configuration file so that “listen=YES” is displayed.

```
[root@GIAC-Linux /]# vi /etc/vsftpd.conf
# Example config file /etc/vsftpd.conf
#
# The default compiled in settings are very paranoid. This sample file
# loosens things up a bit, to make the ftp daemon more usable.
<output omitted>
listen=YES
<output omitted>
```

The line “listen=YES” means that vsftpd will be allowed to run in standalone mode.

The following command will start the vsftpd service in standalone mode.

```
[root@GIAC-Linux vsftpd]# /usr/sbin/vsftpd &
[1] 1769
```

Note that we are still logged in as root and we have not set up any other ftp server. I.e. Port 21 is free for our use with vsftpd.

To verify that vsftpd is running we will use the “ftp” command to connect to the service.

```
[root@GIAC-Linux root]# ftp localhost
Connected to localhost (127.0.0.1).
220 (vsFTPd 1.1.3)
Name (localhost:root): rickywald
331 Please specify the password.
Password:
230 Login successful. Have fun.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Notice that we used localhost (i.e. 127.0.0.1), which is the GIAC-Linux box that we are using.

Running vsftpd at startup.

Note that in our example we shall be using xinetd to start vsftpd so do not make these changes, however for completeness we include these instructions here.

To start vsftpd on startup: First we shall check at which run levels vsftpd is started.

```
[root@GIAC-Linux /]# chkconfig --list vsftpd
vsftpd          0:off   1:off   2:off   3:off   4:off   5:off   6:off
```

Next we shall use the chkconfig command with the --level tag to make vsftpd start in run levels 2, 3, 4 and 5.

```
[root@GIAC-Linux /]# chkconfig --level 2345 vsftpd on
```

Finally we shall check the change has been made with the chkconfig command, but with the --list command.

```
[root@GIAC-Linux /]# chkconfig --list vsftpd
vsftpd          0:off   1:off   2:on    3:on    4:on    5:on    6:off
[root@GIAC-Linux /]#
```

Utilize xinetd to start vsftpd instead of standalone.

Now we will utilize xinetd to start vsftpd on system startup.

The first step is to find the file vsftpd.xinetd.

```
[root@GIAC-Linux /]# locate vsftpd.xinetd
```

/usr/share/doc/vsftpd-1.1.3/vsftpd.xinetd

Next we copy the file vsftpd.xinetd to /etc/xinetd.d/vsftpd.
[root@GIAC-Linux /]# cp /usr/share/doc/vsftpd-1.1.3/vsftpd.xinetd /etc/xinetd.d/vsftpd

Then, open up /etc/xinetd.d/vsftpd using vi and change “disable = yes” to “disable = no”, so that vsftpd running via xinetd is enabled.

```
[root@GIAC-Linux xinetd.d]# vi /etc/xinetd.d/vsftpd
# default: off
# description: The vsftpd FTP server serves FTP connections. It uses \
#       normal, unencrypted usernames and passwords for authentication.
service ftp
{
        socket_type             = stream
        wait                    = no
        user                    = root
        server                  = /usr/sbin/vsftpd
        nice                    = 10
        disable                 = no
        flags                   = IPv4
}
[root@GIAC-Linux xinetd.d]#
```

Earlier we tested vsftpd in standalone mode. It is important to comment out the line:
listen=YES
from /etc/vsftpd.conf in order for xinetd to start vsftpd on demand, rather than with vsftpd running in standalone method.

Restart xinetd as shown below:

```
[root@GIAC-Linux root]# /etc/rc.d/init.d/xinetd restart
Stopping xinetd:                                           [  OK  ]
Starting xinetd:                                           [  OK  ]
[root@GIAC-Linux root]#
```

Verify that Apache, MySQL and VSFTP servers are run at startup

Now reboot the machine. The purpose of this is to verify that Apache (httpd), MySQL Server (mysqld) and FTP server (vsftpd) are ALL started automatically at startup.

View the processes that are running by using the ps command

```
[root@GIAC-Linux root]# ps -ef
UID        PID  PPID  C STIME TTY          TIME CMD
<output omitted>
root       713     1  0 13:34 ?        00:00:00 xinetd -stayalive -reuse -pidfilroot
737     1  0 13:34 ?        00:00:00 /bin/sh /usr/bin/safe_mysqld --dmysql      767
737  0 13:34 ?        00:00:00 [mysqld]
<output omitted>
root       818     1  0 13:34 ?        00:00:00 /usr/sbin/httpd
root       827     1  0 13:34 ?        00:00:00 crond
apache     836   818  0 13:35 ?        00:00:00 [httpd]
apache     837   818  0 13:35 ?        00:00:00 [httpd]
apache     838   818  0 13:35 ?        00:00:00 [httpd]
apache     839   818  0 13:35 ?        00:00:00 [httpd]
apache     840   818  0 13:35 ?        00:00:00 [httpd]
apache     841   818  0 13:35 ?        00:00:00 [httpd]
apache     842   818  0 13:35 ?        00:00:00 [httpd]
apache     843   818  0 13:35 ?        00:00:00 [httpd]
<output omitted>
root      1113  1111  0 13:38 pts/0    00:00:00 bash
root      1139  1113  0 13:38 pts/0    00:00:00 ps -ef
[root@GIAC-Linux root]#
```

Because vsftpd is started on demand by xinetd, type “ftp localhost” to validate that vsftpd is running.

```
[root@GIAC-Linux /]# ftp localhost
Connected to localhost (127.0.0.1).
220 (vsFTPd 1.1.3)
Name (localhost:root): rickywald
331 Please specify the password.
Password:
230 Login successful. Have fun.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Section 5 – CIS Scan and Nessus

CIS Scan for security base lining.

Download and Install CIS Scan

Go to www.cisecurity.org and click the links for CIS Security Benchmarks and Scoring Tools for Linux Level 1. Three downloads will be available:

- LinuxBenchmark.pdf. This itself is a hardening guide to reach the CIS security standard of level 1. We will be using the CIS tools as a monitoring mechanism to measure the security of our system before and after the main bulk of hardening. Download this PDF document onto another machine that has a pdf viewer installed.
- CISscan-1.4.2-1.0.i386.rpm. This is the tool that we will be using to provide an indication as to the security of our system.
- README. We will download the README file.

On the system we download the files to /securitytools/CIS as displayed in the output below. Obviously any location could be used, but by specifying a location we assist with communication within this document.

```
[root@GIAC-Linux CIS]# pwd
/securitytools/CIS
```

One of the downloads available was “Download the MD5 Checksum for the Linux Tool archive”. Click this link to see the MD5 Checksum for the file cis-linux.tar.gz. The output is as follows:

```
624304dcfcfd238723d40606209f502c  cis-linux.tar.gz
```

Next determine the MD5 hash value of the cis-linux.tar.gz file that we just downloaded. I.e.

```
[root@GIAC-Linux CIS]# md5sum cis-linux.tar.gz
624304dcfcfd238723d40606209f502c  cis-linux.tar.gz
```

Now compare the MD5 hash values given from the CIS Security website to that calculated on the host system GIAC-Linux. You will notice that these are the same (both being 624304dcfcfd238723d40606209f502c) hence we will trust the integrity of the file cis-linux.tar.gz that is on our hard disk. This is to mitigate a man in the middle attack where the version of cis-linux.tar.gz has been modified from the original. By modified some Trojan software could have been included or the software could have been modified to give an artificially high security rating, etc.

Next we will unzip, or rather gunzip the file cis-linux.tar.gz
[root@GIAC-Linux CIS]# gunzip /securitytools/CIS/cis-linux.tar.gz

Now we will extract all the files from the tar archive cis-linux.tar using the tar command with the x tag.

[root@GIAC-Linux CIS]# tar xf /securitytools/CIS/cis-linux.tar

Next we view the README file for CIS Scan.

[root@GIAC-Linux cis]# more /securitytools/CIS/cis/README
<ouput omitted>
To install:
rpm -U CISscan-1.4.2-1.0.i386.rpm
To run:
/usr/local/CIS/cis-scan
<output omitted>

As can be seen above, this README file contains instructions on installing and running CIS Scan which we will follow.

[root@GIAC-Linux cis]# rpm -U CISscan-1.4.2-1.0.i386.rpm

Running CIS Scan

Enter the following command to run CIS Scan.

[root@GIAC-Linux cis]# /usr/local/CIS/cis-scan

```
*************************************************************
********** CIS Security Benchmark Checker v1.4.2 **********
*                                                           *
* Lead Developer                : Jay Beale                 *
* Benchmark Coordinator and Gadfly        : Hal Pomeranz    *
*                                                           *
* Copyright 2001 - 2003 The Center for Internet Security  www.cisecurity.org *
*                                                           *
* Please send feedback to linux-scan@cisecurity.org.        *
*************************************************************
        Investigating system...this will take a few minutes...

                        ******
```

Now a final check for non-standard world-writable files, Set-UID and Set-GID programs -- this can take a whole lot of time if you have a large file system.

Your score if there are no extra world-writable files or SUID/SGID programs found will be 4.92 / 10.00 . If there are extra SUID/SGID programs or world-writable files, your score could be as low as 4.62 / 10.00 .

You can hit CTRL-C at any time to stop at this remaining step.

The preliminary log can be found at: /usr/local/CIS/cis-most-recent-log

******

Rating = 4.92 / 10.00

****

To learn more about the results, do the following:

All results/diagnostics:
more /usr/local/CIS/cis-ruler-log.20031026-14:58:52.1927
Positive Results Only:
egrep "^Positive" /usr/local/CIS/cis-ruler-log.20031026-14:58:52.1927
Negative Results Only:
egrep "^Negative" /usr/local/CIS/cis-ruler-log.20031026-14:58:52.1927

For each item that you score or fail to score on, please reference the corresponding item in the CIS Benchmark Document.

For additional instructions/support, please reference the CIS web page:

http://www.cisecurity.org

****

[root@GIAC-Linux cis]#

A CIS Scan rating of 4.92 for GIAC-Linux is poor. You can view the log details as described above in the output from the CIS Scan.

Nessus

Nessus is a scanner/vulnerability assessment tool that we will use to audit our system. We will compare the output before and after hardening.

Installing GCC

Note that GCC is a prerequisite for the installation of Nessus. If GCC is not installed you would receive the following error.

```
[root@GIAC-Linux Nessus]# sh nessus-installer.sh
No compiler found in your $PATH. Can't continue.
If you have no compiler, then get gcc
```

Install GCC.

You can follow the instructions below to install GCC using the RPM from the Red Hat Linux 9 installation CDs. Alternatively GCC would have been installed by selecting Development Tools when at the Package Group Selection screen as GCC is one of the Standard packages in Development Tools.

```
[root@GIAC-Linux /]# mkdir /gcc
[root@GIAC-Linux /]# chmod 700 /gcc
Insert Red Hat Linux 9 CD2 of 3.
You may need to mount the cd with the following command:
[root@GIAC-Linux /]# mount /dev/cdrom
[root@GIAC-Linux /]# cp /mnt/cdrom/RedHat/RPMS/gcc-3.2.2-5.i386.rpm /gcc
```
[Note that we could install gcc straight from the CD, however in this example we copy the rpm file to the hard disk first so that it is readily available in the future in case a reinstall is required.]
```
[root@GIAC-Linux /]# rpm -i /gcc/gcc-3.2.2-5.i386.rpm
```

Obtaining GLIB & GTK

GLIB & GTK are prerequisites for the Nessus GUI.

```
[root@GIAC-Linux /]# ftp ftp.gimp.org
Connected to ftp.gimp.org (128.32.112.248).
220 Welcome to the GIMP.ORG FTP service.
Name (ftp.gimp.org:root): anonymous
331 Please specify the password.
Password:
230 Login successful. Have fun.
Remote system type is UNIX.
Using binary mode to transfer files.
```

We turn on hashing so that we can view the data being transferred.

```
ftp> hash
Hash mark printing on (1024 bytes/hash mark).
```

Next we will move to the correct location where glib and gtk are located.

```
ftp> ls
227 Entering Passive Mode (128,32,112,248,71,220)
<output omitted> drwxrwsr-x   11 0        102          1024 Feb 25  2003 pub
<output omitted>
```

```
ftp> cd pub
250 Directory successfully changed.
ftp> ls
227 Entering Passive Mode (128,32,112,248,32,210)
<output omitted>
drwxrwsr-x   24 1000     102          1024 Oct 24 21:55 gtk
<output omitted>
ftp> ls
227 Entering Passive Mode (128,32,112,248,68,3)
<output omitted>
ftp> cd gtk <output omitted>
ftp> ls
drwxrwsr-x    4 1000     102          2048 Apr 22  2001 v1.2

ftp> cd v1.2
250 Directory successfully changed.
ftp> ls
227 Entering Passive Mode (128,32,112,248,114,56)
150 Here comes the directory listing.
-rw-rw-r--    1 1001     102             0 Apr 22  2001 LATEST-GLIB-1.2.10
-rw-rw-r--    1 1001     102             0 Apr 22  2001 LATEST-GTK+-1.2.10
<output omitted>
```

Now we are at the correct location, we will ftp over glib.

```
ftp> get glib-1.2.10.tar.gz
local: glib-1.2.10.tar.gz remote: glib-1.2.10.tar.gz
227 Entering Passive Mode (128,32,112,248,194,4)
150 Opening BINARY mode data connection for glib-1.2.10.tar.gz (421480
bytes).
######<output omitted>##
226 File send OK.
421480 bytes received in 1.82 secs (2.3e+02 Kbytes/sec)
```

Next we ftp over gtk.

```
ftp> get gtk+-1.2.10.tar.gz
local: gtk+-1.2.10.tar.gz remote: gtk+-1.2.10.tar.gz
227 Entering Passive Mode (128,32,112,248,204,210)
150 Opening BINARY mode data connection for gtk+-1.2.10.tar.gz (2868322
bytes).
###########<output omitted>#########
226 File send OK.
2868322 bytes received in 9.12 secs (3.1e+02 Kbytes/sec)
ftp> quit
221 Goodbye.
```

[root@GIAC-Linux /]#

Installing glib

Note that before installing gtk version 1.2.10, you must have installed a version of glib 1.2.8 or later.

Next use gunzip to unzip " glib-1.2.10.tar.gz" to " glib-1.2.10.tar". Then use tar with the xf tags to extract all the files from " glib-1.2.10.tar" to the "glib-1.2.10" directory which in my case is "/glib/glib-1.2.10". Use the more command to read the "INSTALL" file which contains the instructions on installing glib.

Here is the extract from the INSTALL file with the installation commands.

```
 % cd glib-1.2.10                    # change to the toplevel directory
 % ./configure                       # run the `configure' script
 % make                              # build GLIB
 [ Become root if necessary ]
 % rm -rf /install-prefix/include/glib.h /install-prefix/include/gmodule.h
 % make install                      # install GLIB"
```

Type the above command sequence to complete the installation.

Installing gtk

Next use gunzip to unzip "gtk+-1.2.10.tar.gz" to "gtk+-1.2.10.tar". Then use tar with the xf tags to extract all the files from "gtk+-1.2.10.tar" to the "./gtk+-1.2.10 directory" which in my case is /gtk/gtk+-1.2.10".

```
[root@GIAC-Linux /]# mkdir /gtk
[root@GIAC-Linux /]# chmod 700 /gtk
[root@GIAC-Linux /]# mv gtk+-1.2.10.tar.gz ./gtk

 [root@GIAC-Linux gtk]# gunzip gtk+-1.2.10.tar.gz
 [root@GIAC-Linux gtk]# tar xf gtk+-1.2.10.tar
```

Use the more command to read the "INSTALL" file which contains the instructions on installing gtk.

```
[root@GIAC-Linux gtk]# cd gtk+-1.2.10
[root@GIAC-Linux gtk+-1.2.10]# more ./INSTALL
Prerequisites
=============
GTK+ requires the GLIB library, available at the same location as
you got this package.
```

Simple install procedure

```
=======================

  % gzip -cd gtk+-1.2.10.tar.gz | tar xvf - # unpack the sources
  % cd gtk+-1.2.10                  # change to the toplevel directory
  % ./configure                     # run the `configure' script
  % make                            # build GTK

  [ Become root if necessary ]
  % rm -rf /install-prefix/include/gtk /install-prefix/include/gdk
  % make install                    # install GTK
<output omitted>
```

Now we follow these instructions to install gtk.
[root@GIAC-Linux gtk+-1.2.10]# ./configure
<output omitted>
[root@GIAC-Linux gtk+-1.2.10]# make
<output omitted>
[root@GIAC-Linux gtk+-1.2.10]# rm -rf /install-prefix/include/gtk /install-prefix/include/gdk
[root@GIAC-Linux gtk+-1.2.10]# make install
<output omitted>
make[1]: Leaving directory `/root/gtk/gtk+-1.2.10'
[root@GIAC-Linux gtk+-1.2.10]#

Uudecode(1) / sharutils

Note that uudecode(1) which is part of the sharutils package is a prerequisite for Nessus. If you do not install this package first then you will receive the following error message.
[root@GIAC-Linux Nessus]# sh nessus-installer.sh
The script needs uudecode(1) to run properly
Install the package 'sharutils-*.rpm'

Install the sharutils package as follows.

Insert the Red Hat 9 Linux CD3 of 3.
You may need to mount the CD with the following command:
[root@GIAC-Linux Nessus]# mount /dev/cdrom
[root@GIAC-Linux /]# cd /mnt/cdrom/RedHat/RPMS
[root@GIAC-Linux RPMS]# mkdir /securitytools/Nessus/sharutils
[root@GIAC-Linux RPMS]# chmod 700 /securitytools/Nessus/sharutils
[root@GIAC-Linux RPMS]# cp /mnt/cdrom/RedHat/RPMS/sharutils-*.rpm /securitytools/Nessus/sharutils
[Note that we could install sharutils-* straight from the CD, however in this example we copy the rpm file to the hard disk first so that it is readily available in the future in case a reinstall is required.]
[root@GIAC-Linux Nessus]# cd /securitytools/Nessus/sharutils

[root@GIAC-Linux sharutils]# ls
sharutils-4.2.1-14.i386.rpm

Now install the sharutils rpm.
[root@GIAC-Linux sharutils]# rpm -i ./sharutils-4.2.1-14.i386.rpm
[root@GIAC-Linux sharutils]# cd ..

Nessus Install

Download nessus-2.0.9/nessus-installer from www.nessus.org into /securitytools/Nessus. Obviously any location can be used, however we choose /securitytools/Nessus to aid communication within this document.

View the Nessus README file.

[root@GIAC-Linux Nessus]# more /securitytools/Nessus/README.txt
<output omitted>
Nessus now comes with an easy-to-use installer (nessus-installer.sh).
To use it, just download it and type:
sh nessus-install.sh <output omitted>

Follow the instructions in the README file by using the Nessus installer script to install Nessus.

[root@GIAC-Linux Nessus]# sh nessus-installer.sh

Nessus will run through the installation process.

You will be prompted to answer:
“Where do you want the whole Nessus package to be installed ?
[/usr/local]”
Hit <Enter> to use this default installation location.

Hit Enter when prompted to continue through the installation.

During the installation compiling, we were asked:
/usr/local/lib is not in /etc/ld.so.conf - shall I add it ? [y]
Hit <Enter>.

Eventually you will receive the following output:

“Congratulations ! Nessus is now installed on this host

. Create a nessusd certificate using /usr/local/sbin/nessus-mkcert
. Add a nessusd user use /usr/local/sbin/nessus-adduser

. Start the Nessus daemon (nessusd) use /usr/local/sbin/nessusd -D
. Start the Nessus client (nessus) use /usr/local/bin/nessus
. To uninstall Nessus, use /usr/local/sbin/uninstall-nessus

. Remember to invoke 'nessus-update-plugins' periodically to update your list of plugins
. A step by step demo of Nessus is available at :
http://www.nessus.org/demo/

Press ENTER to quit"

First we shall create the certificate for the Nessus daemon.

```
[root@GIAC-Linux Nessus]# cd /usr/local/sbin
[root@GIAC-Linux sbin]# ls
nessus-adduser  nessus-mkcert  nessus-update-plugins
nessusd         nessus-rmuser  uninstall-nessus
[root@GIAC-Linux sbin]# /usr/local/sbin/nessus-mkcert
```

This script will now ask you the relevant information to create the SSL certificate of Nessus. Note that this information will *NOT* be sent to anybody (everything stays local), but anyone with the ability to connect to your Nessus daemon will be able to retrieve this information.

```
CA certificate life time in days [1460]: 365
Server certificate life time in days [365]:
Your country (two letter code) [FR]: GB
Your state or province name [none]: London
Your location (e.g. town) [Paris]: London
Your organization [Nessus Users United]: GIAC-Example
```

You are informed that the root Certificate Authority and the Nessus server public/private key pairs were created and the locations of the public key (or more accurately the Certificate containing the public key) and private key are given.

The output is as follows:

"Congratulations. Your server certificate was properly created.

/usr/local/etc/nessus/nessusd.conf updated

The following files were created :

. Certification authority :

Certificate = /usr/local/com/nessus/CA/cacert.pem
Private key = /usr/local/var/nessus/CA/cakey.pem

. Nessus Server :
Certificate = /usr/local/com/nessus/CA/servercert.pem
Private key = /usr/local/var/nessus/CA/serverkey.pem"

Press Enter to quit."

Next add a user to the Nessus application.

[root@GIAC-Linux root]# /usr/local/sbin/nessus-adduser
<output omitted>
Add a new nessusd user

Login : nessus_user_login
Authentication (pass/cert) [pass] :
Login password : password

User rules
---
nessusd has a rules system which allows you to restrict the hosts
that nessus_user_login has the right to test.
<output omitted>
Enter the rules for this user, and hit ctrl-D once you are done : <output omitted>
accept client_ip
default deny

Then as instructed press ctrl-D

Note that accept client_ip means that the user "nessus_user_login" is only allowed to use Nessusd on the host machine. This is important as an extra precaution to stop someone using Nessus on this machine as an attacking point to other systems.

"default deny", means that nessus_user_login cannot use Nessus on any other systems. This is a similar concept to the "deny any any" rule at the end of checkpoint firewall rule sets.

You will be prompted to confirm the information entered.
<output omitted>
Is that ok ? (y/n) [y]

Assuming the information is correct, hit Enter so that the default value “y” is accepted.

Running Nessus

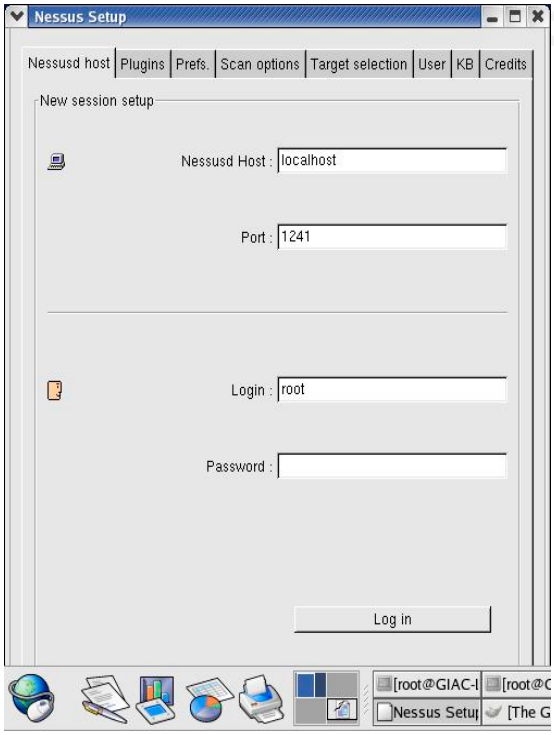
Next we shall start the Nessus Daemon:

[root@GIAC-Linux root]# /usr/local/sbin/nessusd -D

Next we shall start the Nessus Client, which will connect to the Nessus server, instructing the Nessus server to perform scans.

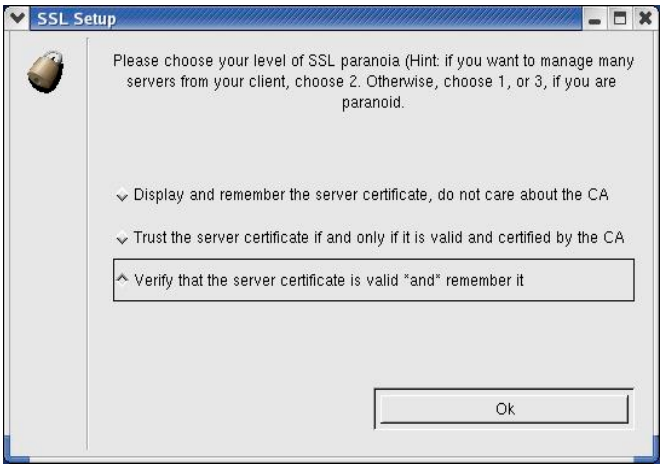
[root@GIAC-Linux root]# /usr/local/bin/nessus

The default nessusd host and port are correct.
Login using the login credentials
Login: nessus_user_login
Password: <Your Password>



Next you are prompted to determine the level of paranoia / acceptable risk when validation of the nessusd certificate. I.e., the server is being authenticating to the client. Currently the GIAC-Linux box is disconnected from the network and all installation files are from trusted sources. Furthermore we have just run through

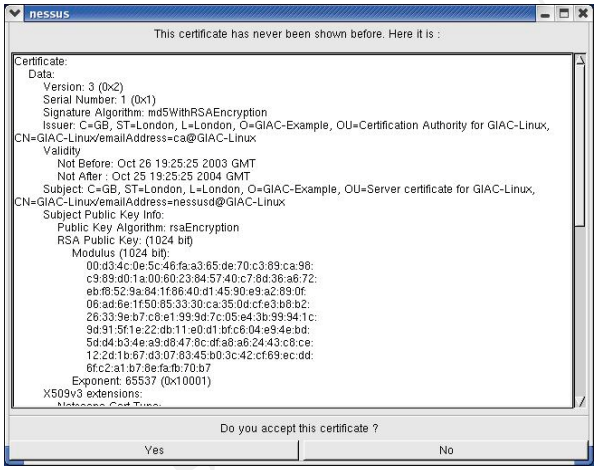
the creation of the nessusd certificate, hence we shall choose the 3rd option as depicted below.



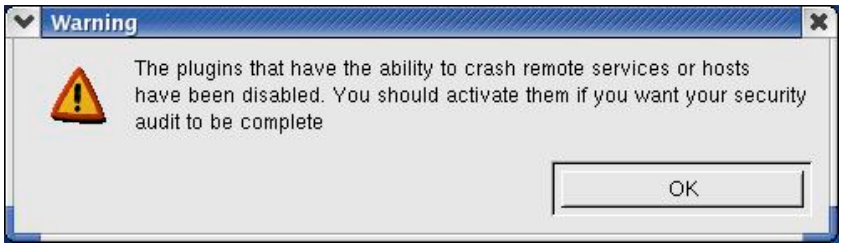
Click Ok.

Next you are presented with the Nessus server certificate. Check the values against what was created. You will note that these match.

Click Yes.

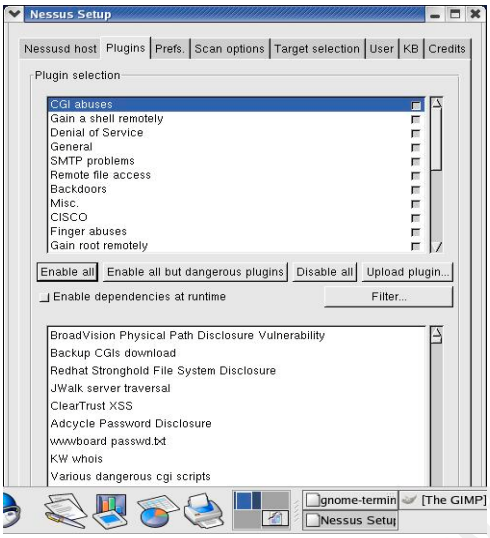


You are informed that by default the dangerous plugins are disabled. Whether you enable or disable the dangerous is a trade off between assuming the risk that the system could be brought down versus getting a more complete audit/scan of the system.



Click OK.

On the plugins tab we will click “Enable all”.



ASIDE: Troubleshooting “start the scan” missing.

Note that when the author reached this point, the “start the scan” button was not displayed, which made the graphical use of Nessus more difficult. Clicking the Red Hat Icon on the desktop, then clicking system settings, then Display, then changing the resolution did not work. The fix was to:

Open the file XF86Config using vi
[root@GIAC-Linux root]# vi /etc/X11/XF86Config

To the following extract, add “1024x768”.

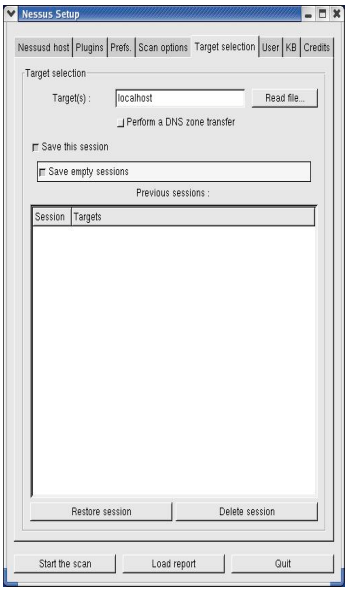
```
Section "Screen"
        Identifier "Screen0"
        Device     "Videocard0"
        Monitor    "Monitor0"
        DefaultDepth     24
        SubSection "Display"
                Depth     24
```

```
        Modes    "1024x768" "800x600" "640x480"
    EndSubSection
```

Save the changes. Then log off and login again.

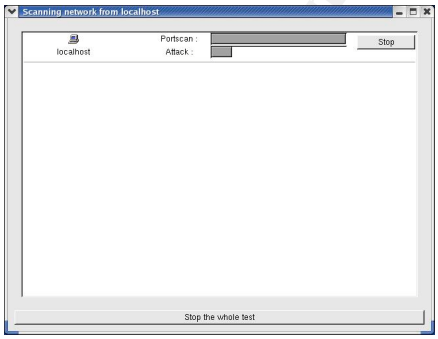
Start the Nessus scan.

In the target selection, correctly enter localhost. This is the machine (i.e. GIAC-Linux) which we are going to scan.



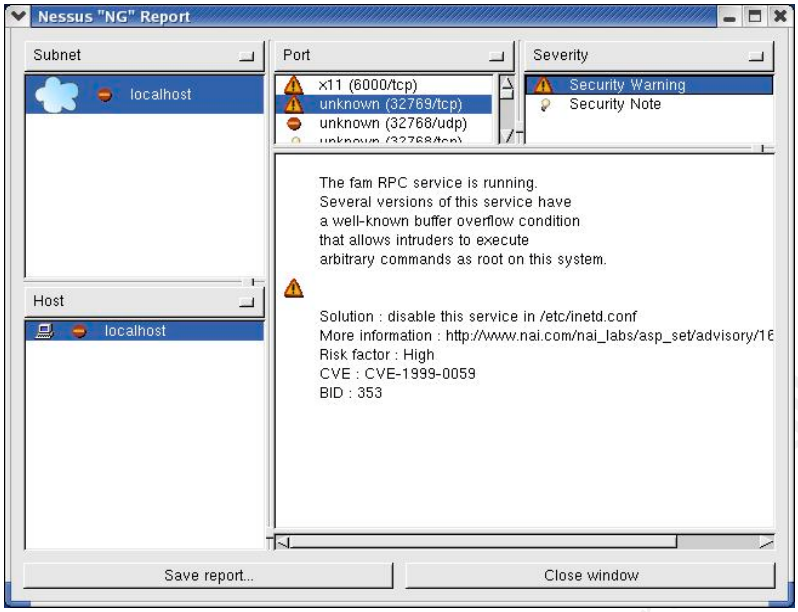
Click “Start the scan”.

Progress bars will be displayed as the scan is performed.



This can take some time.

Highlight the items analogously to the screen shot below to display the results of the scan.



The output of this scan will be used as a baseline for comparison for after full hardening has been performed and for ongoing maintenance.

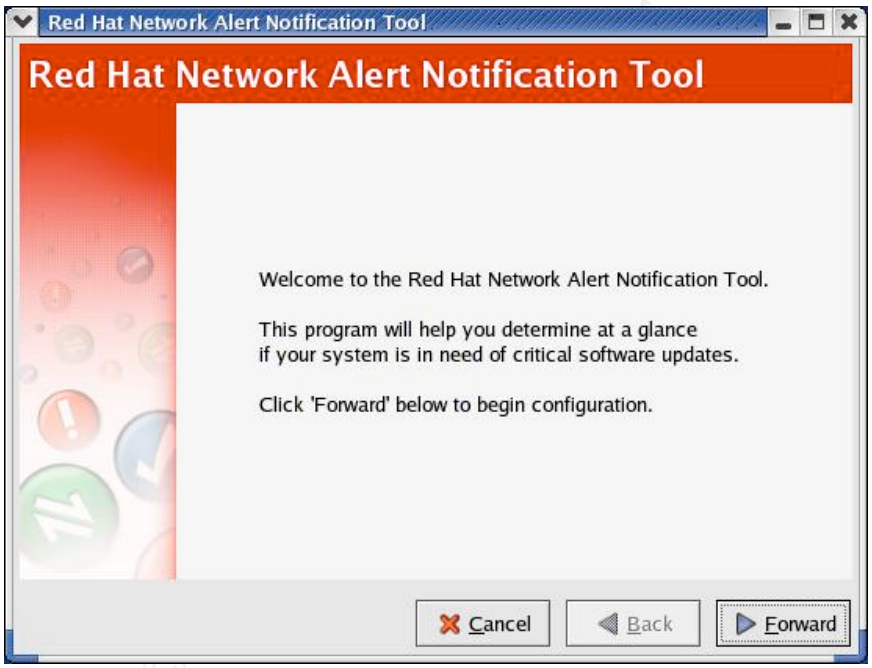
Section 6 - Installing patches from Red Hat Linux

Next we are going to install the patches from Red Hat Linux. Notice the red circle with the flickering white exclamation mark.

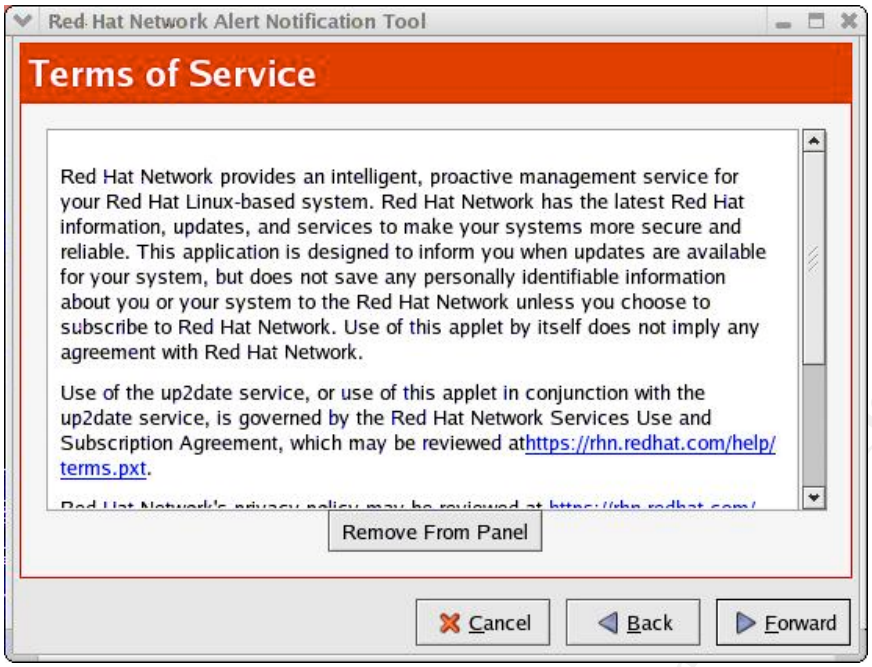
As described at http://www.redhat.com/archives/rhl-beta-list/2003-September/msg00099.html, copy the rhn.redhat.com keys found at https://rhn.redhat.com/help/RHNS-CA-CERT to /usr/share/rhn. This is because "One of the rhn.redhat.com site keys expired".

Configuration of the Red Hat Network Alert Notification Tool

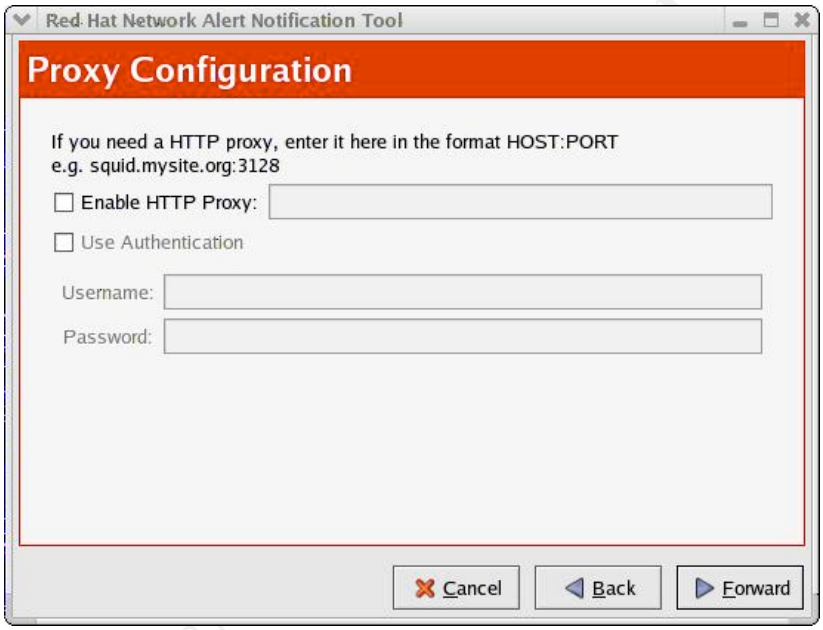
Double click the red circular icon with the left mouse button.



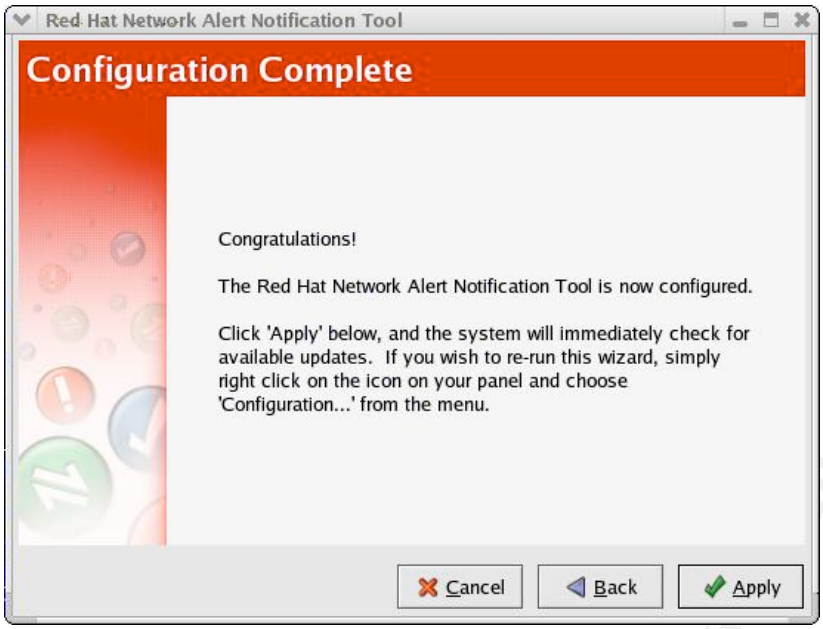
Click Forward.



Click Forward



In this example a proxy is not being used so we simply click forward. Please configure your proxy as appropriate for your environment.



You are informed that the Red Hat Alert Notification Tool is configured.

Click “Apply”.

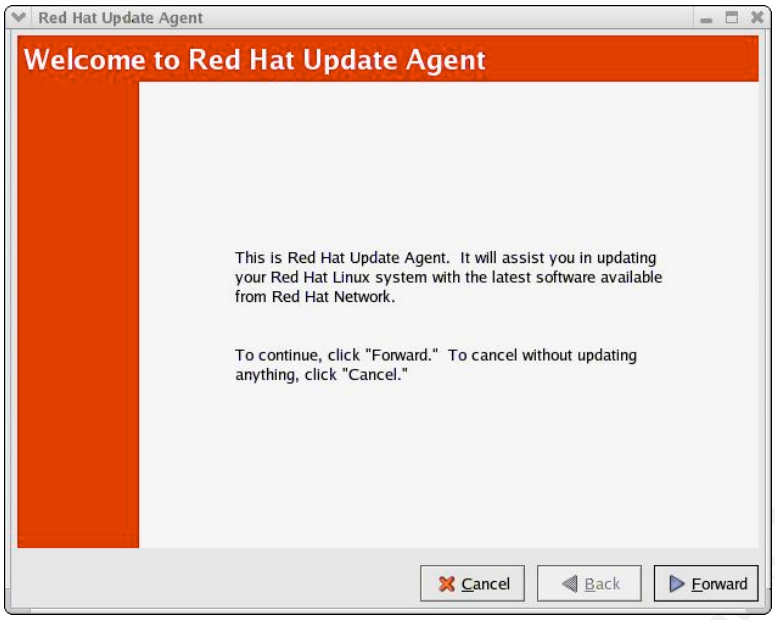
Notice that the task bar update icon has changed to green.

Registering the System Profile

Double click the green icon.

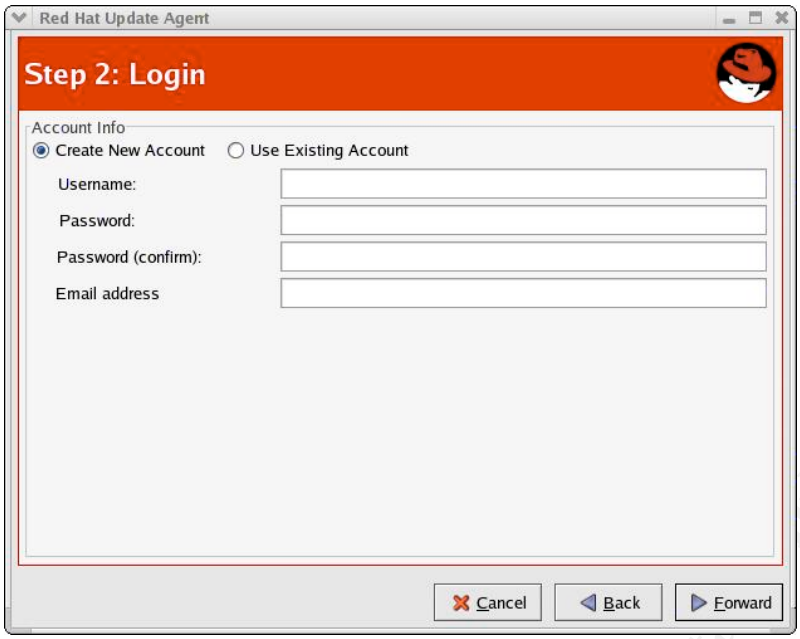
Because this system is not yet registered with the Red Hat Network, we are given the above prompt.

Click Register with RHN.



Click Forward.

Review the privacy statement then click Forward.

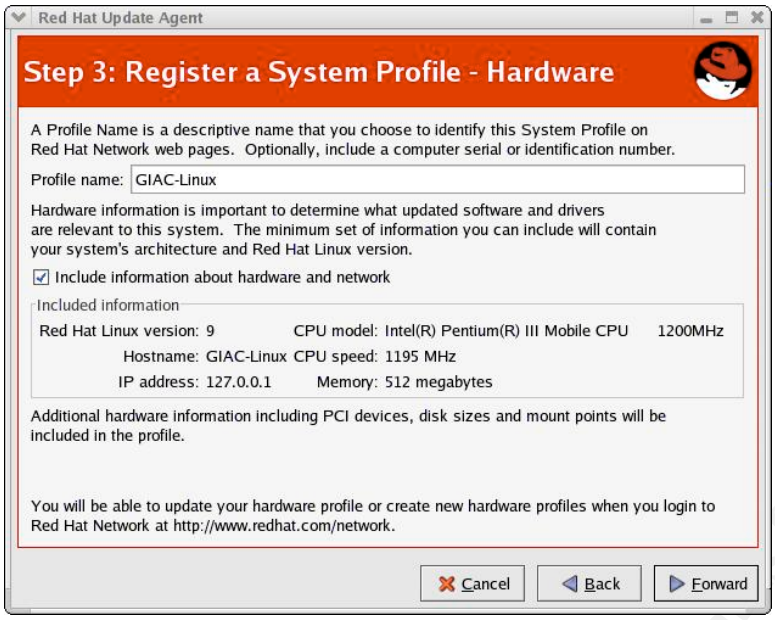


Assuming that we do not already have an existing account that we can use, enter a username, password, confirmation password and email address.

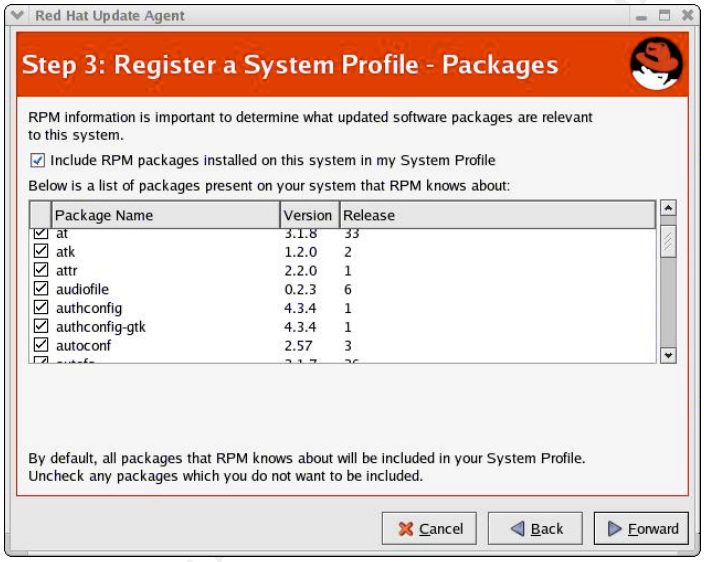
Click Forward.

Enter your user information.

Click Forward.

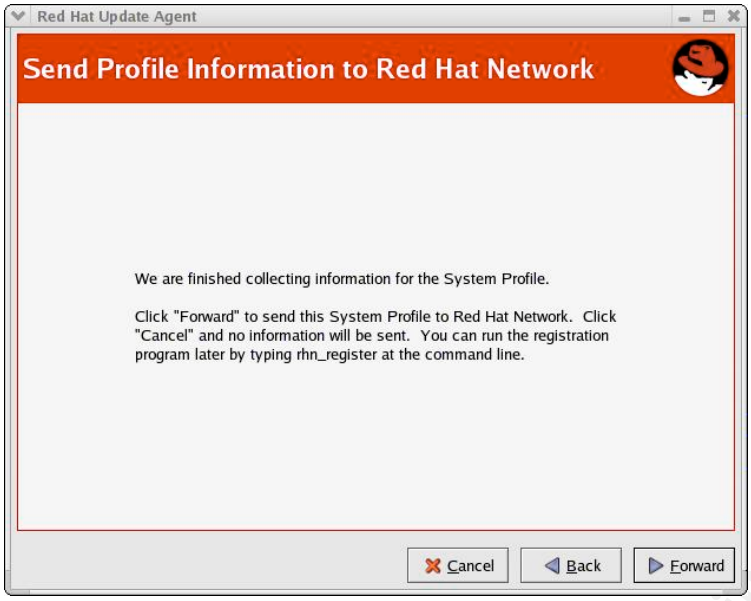


Next in the register system profile – Hardware screen, enter the hostname of the system, which in this case is GIAC-Linux. Check the hardware information and click Forward.

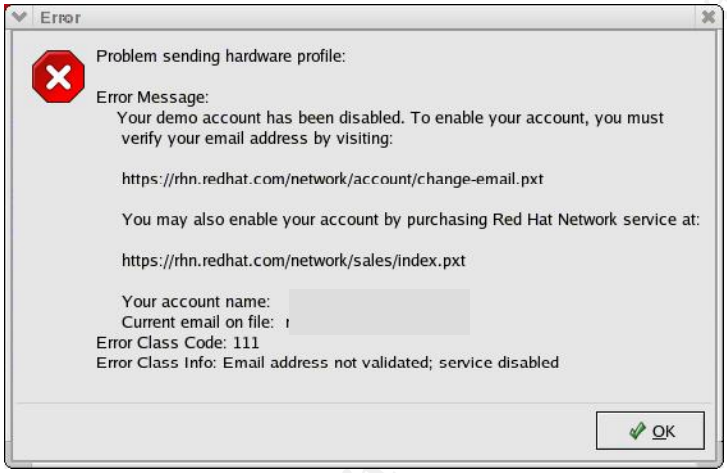


Ensure that the “Include RPM packages installed on this system in my System Profile” is checked. Leave all packages checked.

Click Forward.



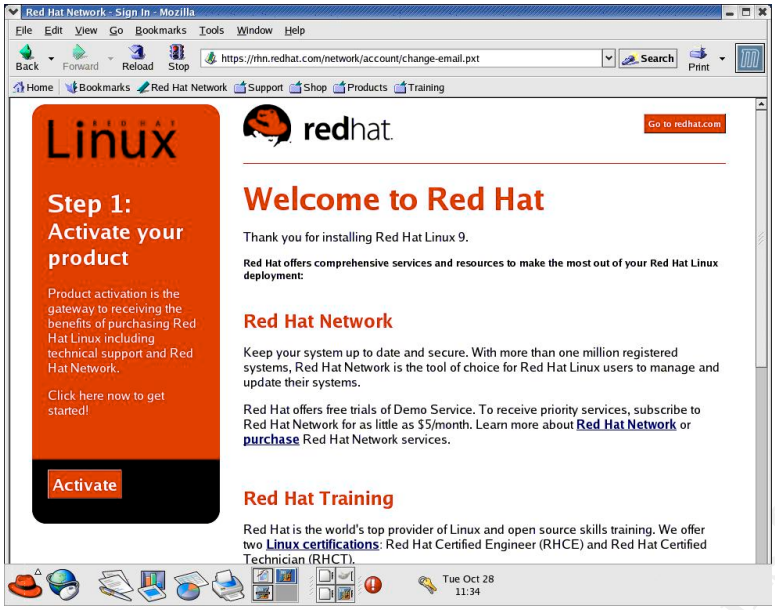
Click Forward.



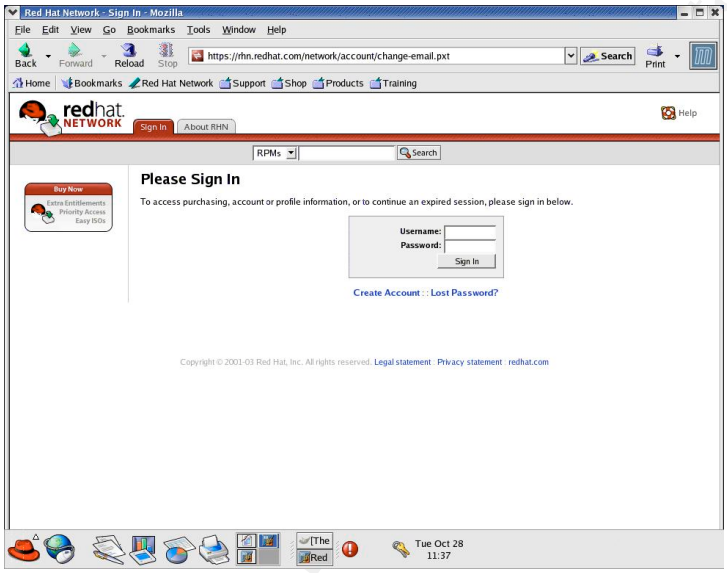
You receive the above error message, which prompts you to verify your email address with Red Hat.

Verifying your email address with Red Hat.

Open Mozilla and go to https://rhn.redhat.com/network/account/change-email.pxt

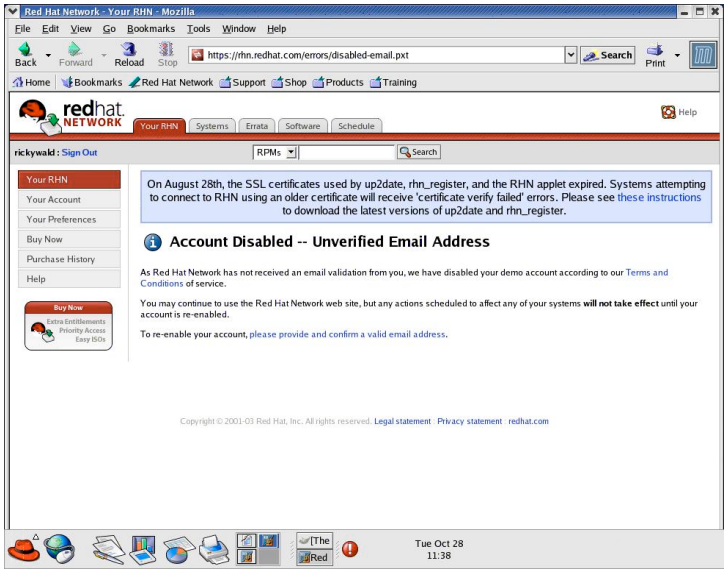


Click Activate.

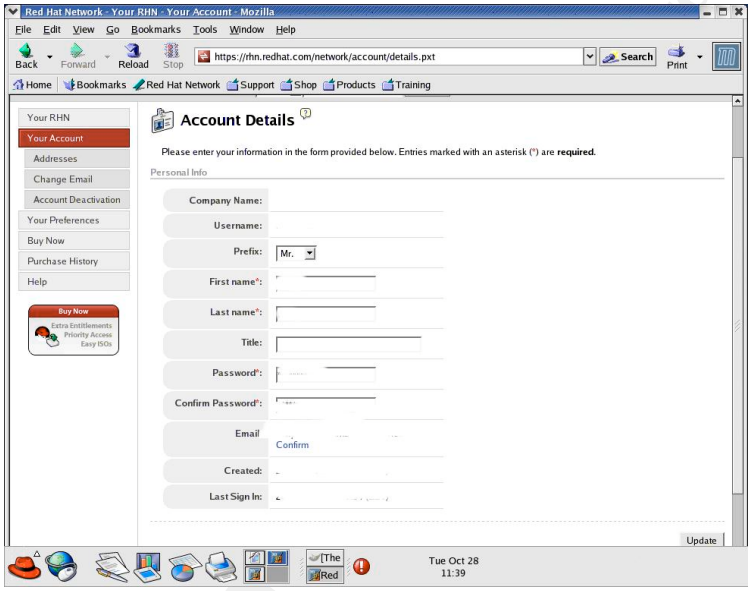


Enter the username and password that we set earlier in step 2, Login.

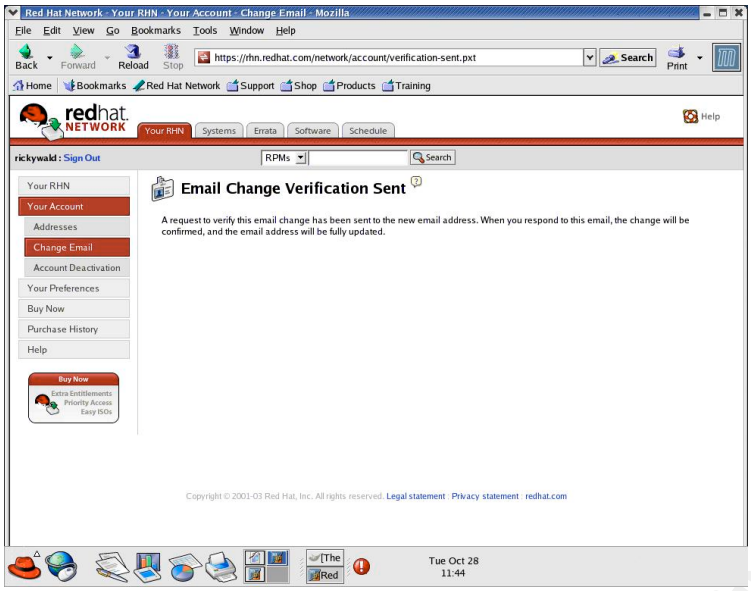
Click Sign in.



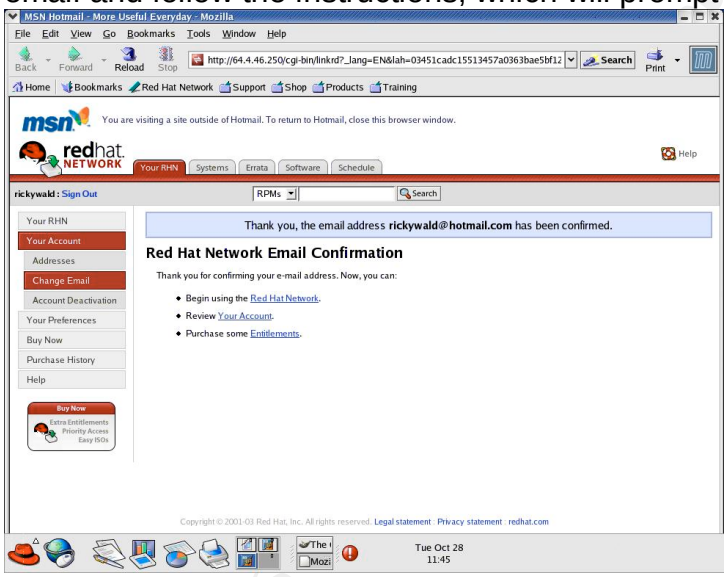
Click “please provide and confirm a valid email address”.



Enter your information. Click Update.



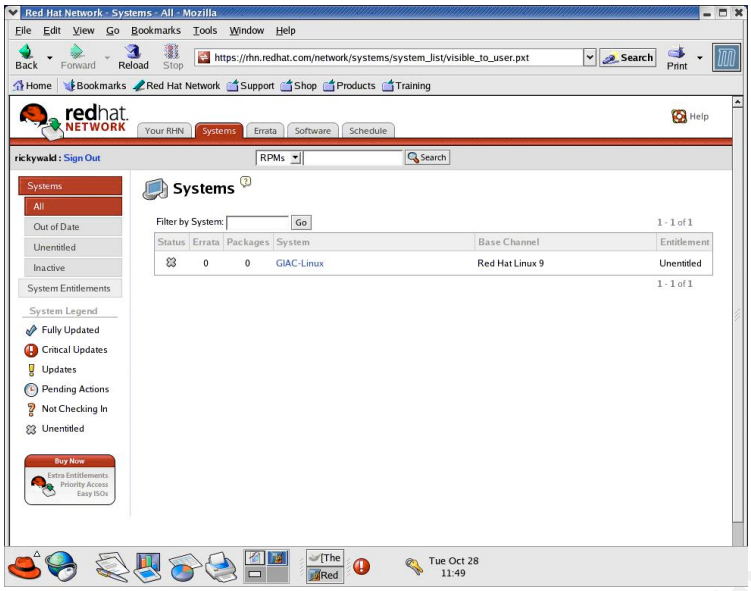
You are informed that a verification email has been sent to you. Check your email and follow the instructions, which will prompt you to click on a URL link.



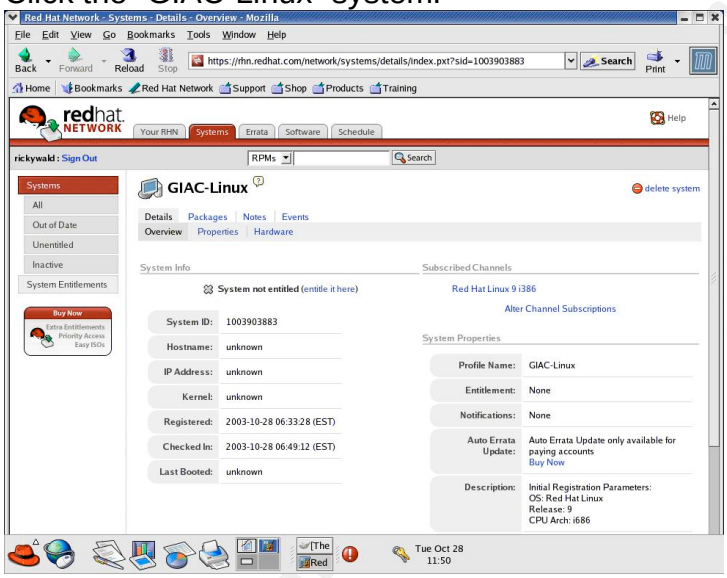
Once you go to this URL, you will be informed that your email address has been confirmed.

Setting the entitlement

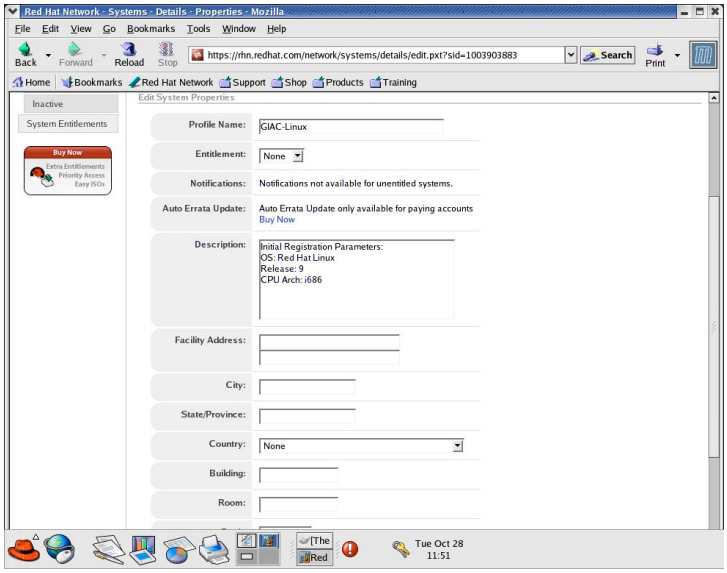
Click the systems tab.



Click the “GIAC-Linux” system.

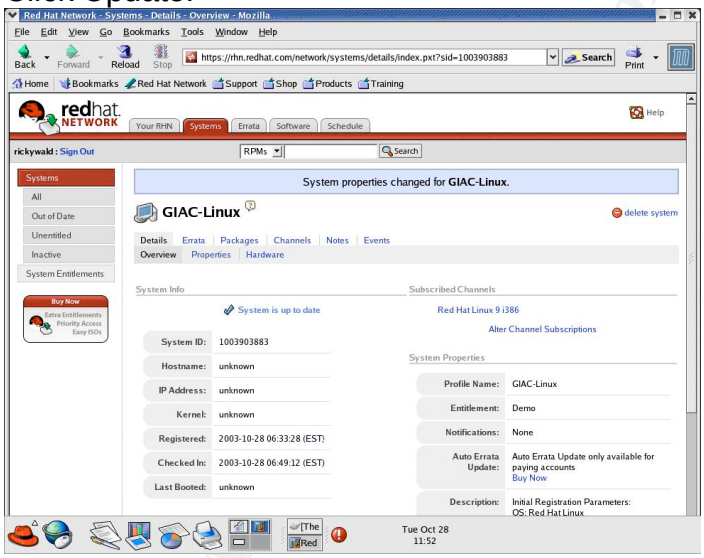


Click “entitle it here”.



Change the entitlement field to the appropriate setting, which for this example is demo.

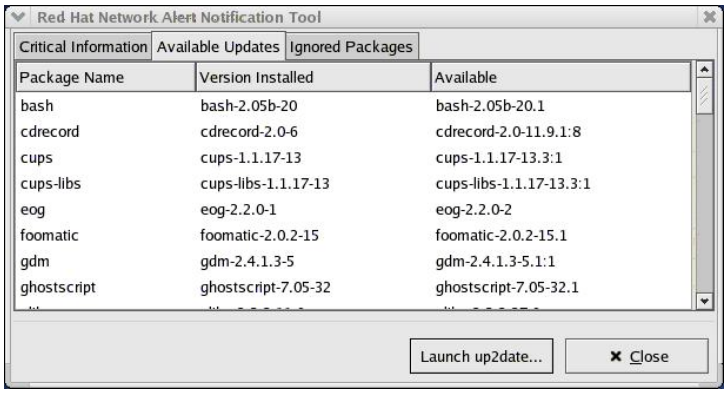
Click Update.



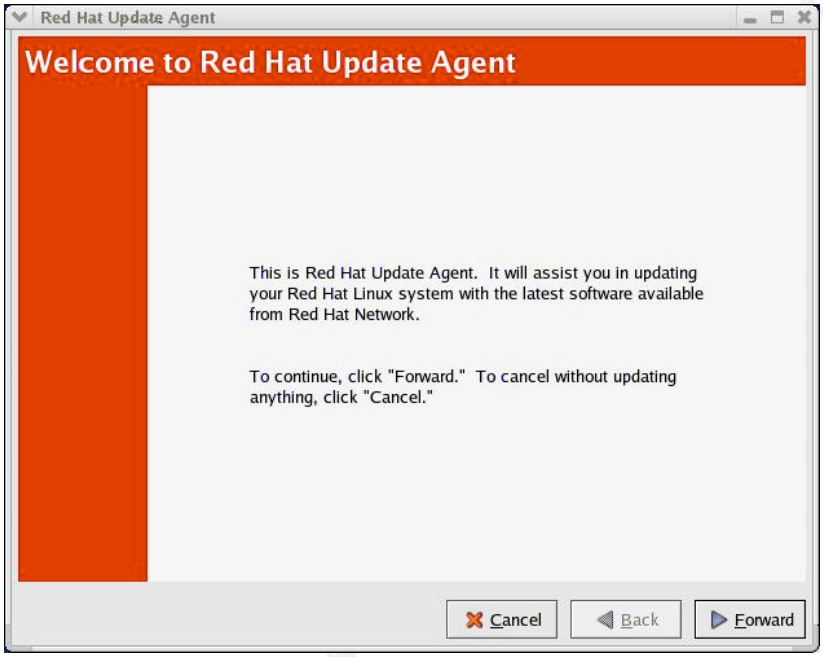
You are informed that the system is up to date.

Next double click on the red circular update icon with the left mouse button.

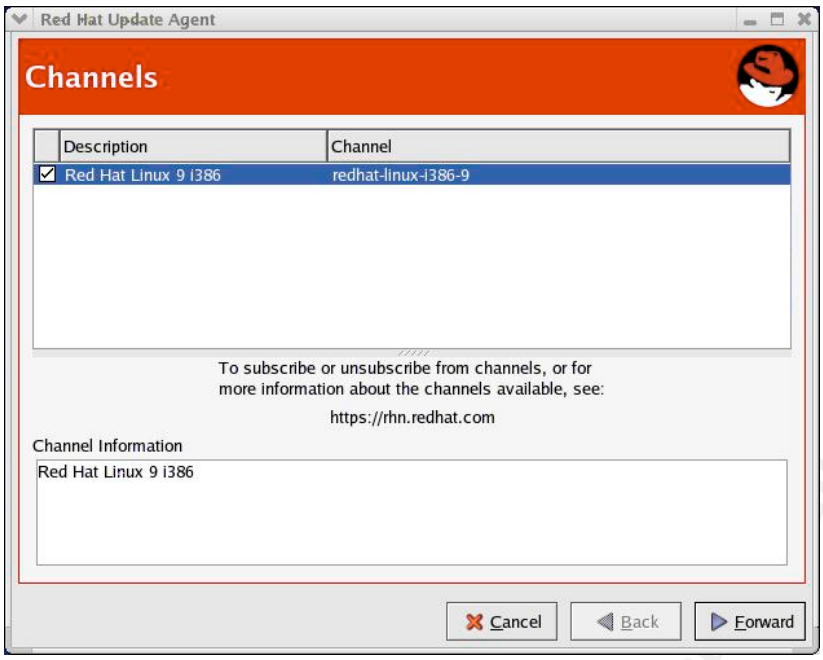
Retrieving and installing packages with up2date



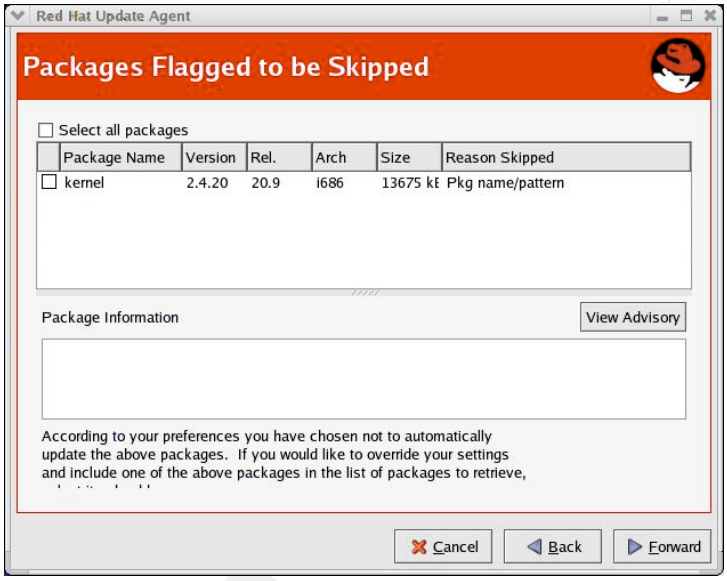
The Red Hat Network Alert Notification Tool screen is displayed. Click “Launch up2date”.



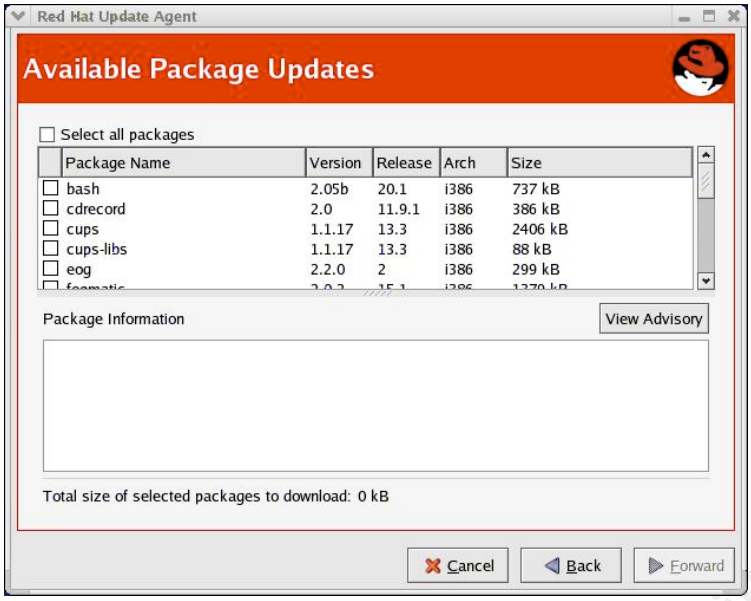
Click Forward.



Click Forward.

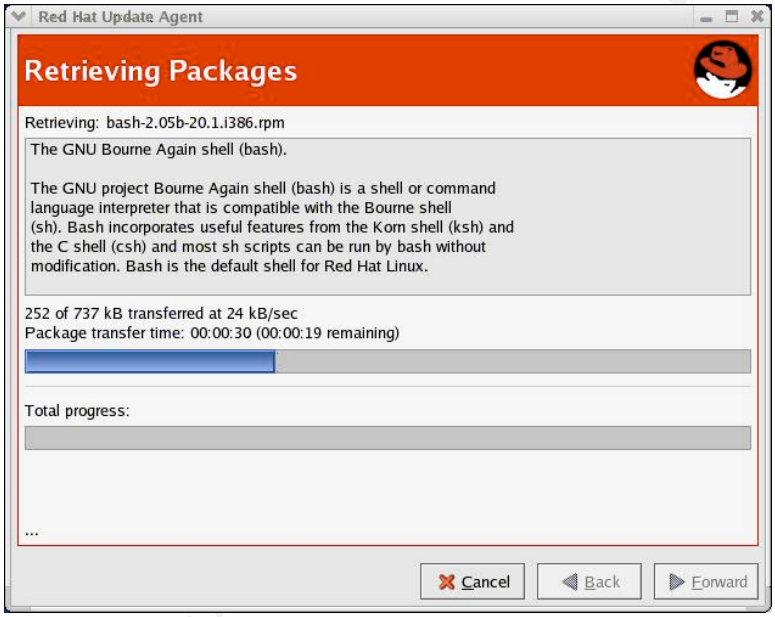


Click Forward.

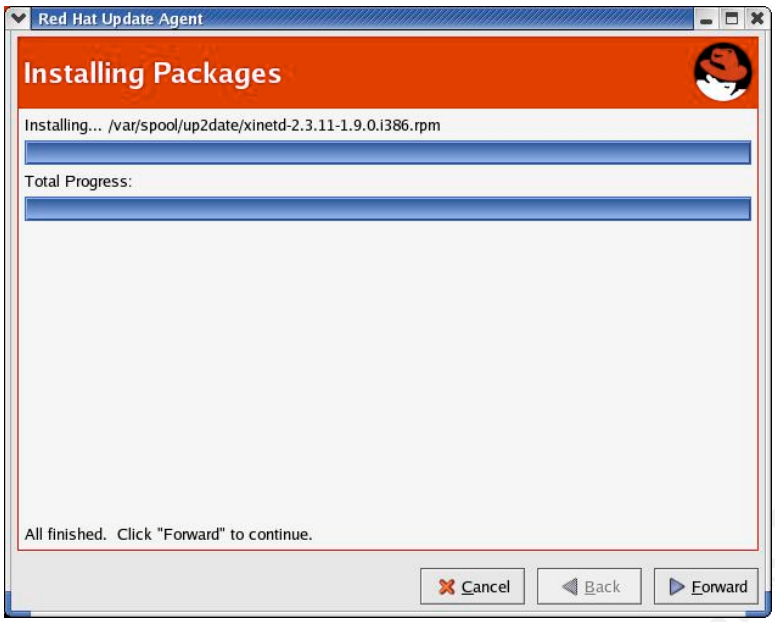


Click the “Select all packages” checkbox.

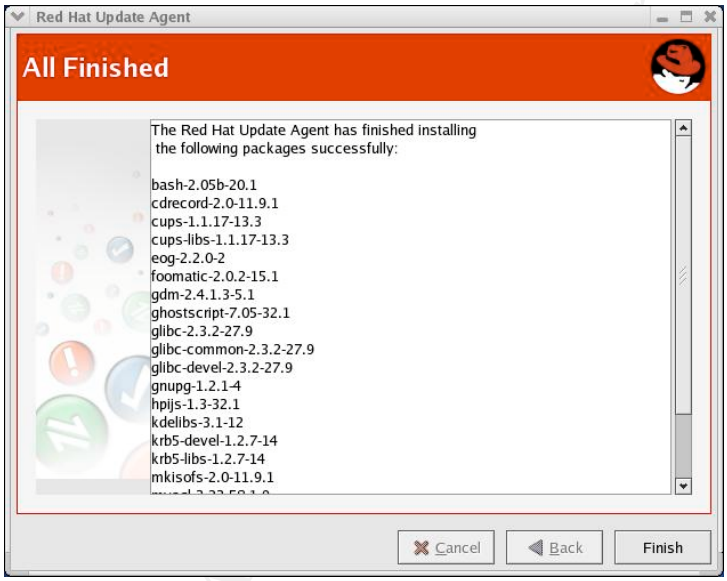
Click Forward.



Once the packages have been retrieved, click Forward.



Once the packages have been installed, click Forward.



You are informed that the packages have been installed. Click Finish.

Section 7 – Further Hardening.

Message of the Day and keep out messages

Message of the Day: We will now add a “keep out” message to the Message of the Day, by modifying the /etc/motd file.

[root@GIAC-Linux etc]# echo Unauthorized access, or attempts to gain unauthorized access to this site is prohibited and trespassers will be prosecuted >> /etc/motd

If you want, you can add a “keep out” message for x windows by modifying /etc/X11/gdm/gdm.conf with vi.
Comment out the line “Greeter=/usr/bin/gdmgreeter”
Change “RemoteGreeter=/usr/bin/gdmlogin” to “Greeter=/usr/bin/gdmlogin”.
Then add your text between “Welcome” and “%n”.
To verify this, simply reboot the machine and see the login screen.

TCP Wrappers

The /usr/sbin/tcpd server determines which hosts can access which INET services. I.e. /usr/sbin/tcpd can restrict access to specific services based on the host, such that there can be segregation between which hosts can access which services. Thus John Doe may only be able to access service 1 and Jane Doe may only be allowed to access service 2. In this file it is best to use ip addresses rather than host names to prevent an attacker giving themselves access by modifying the DNS entries. /usr/sbin/tcpd determines this access based on two files called /etc/hosts.allow and /etc/hosts.deny.

First we will open /etc/hosts.allow using vi.
[root@GIAC-Linux etc]# vi /etc/hosts.allow

Add the following then save the changes and quit:

“sshd: 127.0.0.1: ALLOW
sshd: 192.168.3.2: ALLOW
vsftpd: 127.0.0.1: ALLOW
vsftpd: 192.168.1.*: ALLOW”

Next open /etc/hosts.deny using vi.
[root@GIAC-Linux etc]# vi /etc/hosts.deny

Add the following then save the changes and quit:

"ALL: ALL"

To test this these we can:

- Connect over SSL from the host machine and from 192.168.3.2
- Fail to connect over SSL from an IP address other than 192.168.3.2 for example 192.168.3.3
- FTP a file from the host machine and from 192.168.1.x
- Fail to FTP the same file from an IP address other than 192.168.1.x for example 192.168.3.3

Locking down services.

Viewing and Turning off unneeded xinetd services using chkconfig

First view which xinetd services are started by looking in the /etc/xinetd.d directory, then we will do this another way by using the chkconfig command.

```
[root@GIAC-Linux xinetd.d]# ls
chargen     cups-lpd  daytime-udp  echo-udp  servers  sgi_fam  time-udp
chargen-udp  daytime  echo         rsync     services  time     vsftpd

[root@GIAC-Linux xinetd.d]# chkconfig --list
kudzu           0:off   1:off   2:off   3:on    4:on    5:on    6:off
syslog          0:off   1:off   2:on    3:on    4:on    5:on    6:off
netfs           0:off   1:off   2:off   3:on    4:on    5:on    6:off
network         0:off   1:off   2:on    3:on    4:on    5:on    6:off
random          0:off   1:off   2:on    3:on    4:on    5:on    6:off
rawdevices      0:off   1:off   2:off   3:on    4:on    5:on    6:off
pcmcia          0:off   1:off   2:on    3:on    4:on    5:on    6:off
saslauthd       0:off   1:off   2:off   3:off   4:off   5:off   6:off
keytable        0:off   1:on    2:on    3:on    4:on    5:on    6:off
apmd            0:off   1:off   2:on    3:on    4:on    5:on    6:off
atd             0:off   1:off   2:off   3:on    4:on    5:on    6:off
gpm             0:off   1:off   2:on    3:on    4:on    5:on    6:off
autofs          0:off   1:off   2:off   3:on    4:on    5:on    6:off
iptables        0:off   1:off   2:on    3:on    4:on    5:on    6:off
irda            0:off   1:off   2:off   3:off   4:off   5:off   6:off
nscd            0:off   1:off   2:off   3:off   4:off   5:off   6:off
isdn            0:off   1:off   2:on    3:on    4:on    5:on    6:off
sshd            0:off   1:off   2:on    3:on    4:on    5:on    6:off
portmap         0:off   1:off   2:off   3:on    4:on    5:on    6:off
```

```
nfs             0:off   1:off   2:off   3:off   4:off   5:off   6:off
nfslock         0:off   1:off   2:off   3:on    4:on    5:on    6:off
sendmail        0:off   1:off   2:on    3:on    4:on    5:on    6:off
rhnsd           0:off   1:off   2:off   3:on    4:on    5:on    6:off
crond           0:off   1:off   2:on    3:on    4:on    5:on    6:off
anacron         0:off   1:off   2:on    3:on    4:on    5:on    6:off
xfs             0:off   1:off   2:on    3:on    4:on    5:on    6:off
xinetd          0:off   1:off   2:off   3:on    4:on    5:on    6:off
cups            0:off   1:off   2:on    3:on    4:on    5:on    6:off
ntpd            0:off   1:off   2:off   3:off   4:off   5:off   6:off
firstboot       0:off   1:off   2:off   3:off   4:off   5:off   6:off
mysqld          0:off   1:off   2:on    3:on    4:on    5:on    6:off
postgresql      0:off   1:off   2:off   3:off   4:off   5:off   6:off
vsftpd          0:off   1:off   2:on    3:on    4:on    5:on    6:off
xinetd based services:
        chargen-udp:    off
        rsync:  off
        chargen:        off
        daytime-udp:    off
        daytime:        off
        echo-udp:       off
        echo:   off
        services:       off
        servers:        off
        time-udp:       off
        time:   off
        vsftpd: on
        cups-lpd:       off
        sgi_fam:        on
[root@GIAC-Linux xinetd.d]#
```

We will now turn off:
sgi_fam
In fact the candidates to turn off are;
chargen, chargen-udp, cups-lpd, daytime, daytime-udp, echo, echo-udp, eklogin, finger, gssftp, imap, imaps, ipop2, ipop3, krb5-telnet, klogin, kshell, ktalk, ntalk, pop3s, rexec, rlogin, rsh, rsync, servers, services, sgi_fam, talk, telnet, tftp, time, time-udp, wu-ftpd.
[source: LinuxBenchmark.pdf, section 2.1, from CIS]

```
[root@GIAC-Linux xinetd.d]# chkconfig sgi_fam off
```

One method for Viewing and turning off unneeded services that are started in run level 2.

In this section when referring to turn off, it is meant that the service should not start at startup, rather than turning off a running service.

The following is a list of rc2.d boot services that are candidates for being turned off:
S30sysid.net
S71sysid.sys
S72autoinstall
S73cachefs.daemon
S93cacheos.finish
S40llc2
S47pppd
S95ncad
S47asppp
S70uucp
S72slpd
S75flashprom
S80PRESERVE
S89PRESERVE
S85power
S89bdconfig
S90wbem
S94ncalogd
The following is a list of rc3.d boot services that are candidates for being turned off:
S77dmi
S80mipagent
[Source: Company internal Unix Security Presentation.]

Check that none of these services are started:

```
[root@GIAC-Linux rc2.d]# ls /etc/rc2.d
K03rhnsd       K50xinetd  K95firstboot  S20random    S85gpm
K05atd         K72autofs  K95kudzu      S24pcmcia    S85httpd
K05saslauthd   K74nscd    S08iptables   S26apmd      S90crond
K15postgresql  K74ntpd    S09isdn       S55sshd      S90cups
K20nfs         K75netfs   S10network    S60vsftpd    S90xfs
K24irda        K86nfslock S12syslog     S78mysqld    S95anacron
K44rawdevices  K87portmap S17keytable   S80sendmail  S99local
[root@GIAC-Linux rc2.d]# cd /etc/rc3.d
[root@GIAC-Linux rc3.d]# ls
K05saslauthd   S05kudzu    S17keytable  S56rawdevices  S90crond
K15postgresql  S08iptables S20random    S56xinetd      S90cups
K20nfs         S09isdn     S24pcmcia    S60vsftpd      S90xfs
K24irda        S10network  S25netfs     S78mysqld      S95anacron
K74nscd        S12syslog   S26apmd      S80sendmail    S95atd
K74ntpd        S13portmap  S28autofs    S85gpm         S97rhnsd
K95firstboot   S14nfslock  S55sshd      S85httpd       S99local
```

[root@GIAC-Linux rc3.d]#

For each of these boot services, if they were started by the run control scripts, in the corresponding rc#.d directory, we could change the filename by prefixing ".No" or any other identifier, so that there file remains in case we wish to undo the change, but the service would no longer be started upon entering the respective run level.

NTSYSV – Second method for turning off unneeded services.

In this section when referring to turn off, it is meant that the service should not start at startup, rather than turning off a running service.

Now to demonstrate a second method to turn off services, we will use the command "ntsysv" to view which services are running and turn off the remaining services that we do not want started at startup.
[Note that man pages in particular were used to assist in the writing of this section.]

We now list all the services displayed with ntsysv, write a short description of their purpose and state whether we want these services started at startup or not.

Anacron – This can be used to run tasks with a specified frequency with the key differentiator from other common schedulers being that anacron does not rely on the machine being on. The corresponding configuration file is /etc/anacrontab. We will not be using anacron, so turn this off by hitting the space key when the star next to anacron is highlighted.

Apmd – This will be left on for power management, for example by alerting users when the power is low.

Atd – This runs jobs queued by at. "At" runs jobs from the standard input, i.e. The keyboard, when those jobs are supposed to run at a later time. Leave this on as it is a business requirement, and so that we can demonstrate ways to secure this in later sections.

Autofs – This automatically mounts file systems. Turn this off as we will not be accessing remote files via NFS.

Crond – This runs in the background to execute scheduled commands. Leave this on. This is a business requirement.

Cups – This is the common Unix printing system. Leave this on; (business requirement).

gpm – This is used to perform "cutting" and "pasting". Turn this off.

Iptables – This is used to control how Linux handles IP packet filtering. This will be turned on.

Isdn – Turn this off.

Keytable – This loads the selected keyboard map and hence we will require this to be started at boot.

Kudzu – As the man pages state, “detects and configures new and/or changed hardware on a system”. As we do not want users at the console to be able to add/change hardware, this will be left off.

Netfs – This mounts and un-mounts network file systems. In our environment we are not using this functionality, so this will be turned off.

Network – This activates network interfaces, hence leave this on.

nfslock – In our environment we are not using NFS and hence will not be using the NFS file locking service.

PCMCIA – This handles PCMCIA cards and hence will be turned off.

Portmap – This translates Remote Procedure Call's program numbers into DARPA protocol port numbers. As we do not intend to use RPC, this will be turned off.

Random – This is a random number generator, which will be turned off. If required for generating keys etc in the future, this can be turned back on as and when required.

Rawdevices – This is used for the correspondence of raw devices to block devices and will be left running.

Rhnsd – This periodically will check if there are any updates available from the Red Hat Network, hence this will be left running from a maintenance standpoint.

Sendmail – This delivers emails and will be turned off.

Syslog – This will be left on for logging.

Xfs – This “supplies fonts to X Window System display servers”. Although not ideal, this will be left running.

Xinetd - When a connection arrives, xinetd starts the appropriate service. This will be left running. Note that xinetd will manage starting vsftpd.

Vsftp, sshd, syslog and mysqld will be left running.

So, to summarize, the services that will be started will be:

Apmd, Atd, Crond, Cups, iptables, Keytable, Network, Rawdevices, Rhnsd, Xfs, Xinetd, Vsftpd, sshd, syslog and mysqld.

HTTPD File Permissions

First we are going to set the correct Linux permissions on the Apache executable /usr/local/apache2/bin/httpd.

If it is not already, make httpd owned by root.
[root@GIAC-Linux bin]# chown 0 /usr/local/apache2/bin/httpd

If it is not already, make httpd owned by root's group.
[root@GIAC-Linux bin]# chgrp 0 /usr/local/apache2/bin/httpd

Set the permissions on httpd.
[root@GIAC-Linux bin]# chmod 511 /usr/local/apache2/bin/httpd

Verify that the permissions for httpd are correct.
[root@GIAC-Linux bin]# ls -n /usr/local/apache2/bin/httpd
-r-x--x--x 1 0 0 2146022 Oct 27 10:13 /usr/local/apache2/bin/httpd

Next check that /, /usr, /usr/local are only modifiable by root.
[root@GIAC-Linux usr]# ls -n /
[root@GIAC-Linux usr]# ls -n /usr

Securing vsftpd.

Note that, in particular the man pages were used to write this section.

VSFTPD Configuration File

Below are a list of configuration options in /etc/vsftpd.conf that control the behavior of vsftpd. By each item we note the value that we will set for this variable.

anonymous_enable=NO so that anonymous login is not allowed.

Leave “anon_mkdir_write_enable=NO” commented out so that anonymous users cannot create directories.

anon_other_write_enable=NO so that anonymous users cannot perform non-upload/create write operations such as deletion or renaming.

Leave “anon_upload_enable” commented out so that anonymous users cannot upload files, no matter what the conditions. Also set the value to “NO” in case an attack is run which causes certain comment symbols to be ignored. This concept can be applied throughout the hardening.

Leave “async_abor_enable” commented out because the clients accessing this FTP server are not “ill advised”.

Leave “ascii_upload_enable=YES” and “ascii_download_enable=YES” commented out to prevent attacks from using this method to consume I/O resources.

check_shell=YES so /etc/shells will be checked for a valid user shell for local logins.

deny_email_enable=NO and commented out as anonymous access is not allowed, however if this was an anonymous FTP server then this would be left on and set to “NO” so that you can add password email responses for anonymous users to /etc/vsftpd.banned_emails, which will cause them to be denied.

guest_enable=NO so that non-anonymous logins are classed as themselves.

local_enable=YES so that local logins controlled by /etc/passwd are allowed.

local_umask=077

log_ftp_protocol=YES so that all FTP requests and responses are logged. This is beneficial from a security audit perspective.

Leave “ls_recurse_enable” commented out so that “ls -R” cannot be used at the top of a large site to use system resources.

no_anon_password=NO so that anonymous users need to enter a password (if anonymous access were to be turned back on).

nopriv_user=ftpsecure

pasv_promiscuous=NO so that the PASV security check is used.

port_promiscuous=NO so that outgoing data connections can only connect to the client.

setproctitle_enable=NO so that in the system process, vsftpd will be shown, rather than the actual status.

tcp_wrappers=YES so that incoming connections will be controlled by TCP wrappers.

userlist_deny=NO so that users are denied login unless they are explicitly listed in userlist_file.

userlist_enable=YES so that userlist_deny is examined.

xferlog_enable=YES so that both uploads (which should not be allowed) and downloads are logged.

xferlog_std_format=NO so that logs are not transferred to ftpd xferlog format.

max_per_ip=5 so that a machine with a single IP address cannot perform a DOS attack with the number of ftp connections.

ftpd_banner=Unauthorized access, or attempts to gain unauthorized access to this site is prohibited and trespassers will be prosecuted.

log_ftp_protocol=YES. Create this so that all commands sent to the vsftp server are logged.

Type “vi /etc/vsftpd.conf” and make the above changes.

Unprivileged user and access control files (vsftpd.user_list, vsftpd.ftpusers and vsftpd.banned_emails)

Add the unprivileged user called ftpsecure.

[root@GIAC-Linux etc]# adduser ftpsecure

Modify /etc/vsftpd.user_list by adding the line “rickywald”. Remove (or at least comment out) all the system accounts, which are bin, daemon, adm, lp, sync, shutdown, halt, mail, news, uucp, operator, games and nobody. Thus “rickywald” will be allowed, but the system accounts will not be.

Note that the system accounts are, but “rickywald” is NOT listed in /etc/vsftpd.ftpusers (which lists denied users).

As this is not an anonymous FTP server, we will not create the banned email list file. However if this was an anonymous FTP server, then we would create a banned email list as follows.

```
[root@GIAC-Linux etc]# touch /etc/vsftpd.banned_emails
[root@GIAC-Linux etc]# chmod 600 /etc/vsftpd.banned_emails
```

Add any email addresses you require to this file.

Verification of access control.

Restart vsftpd with the following command.

```
[root@GIAC-Linux etc]# service vsftpd restart
```

Verify that anonymous users cannot login and that the banner is correctly displayed.

```
[root@GIAC-Linux etc]# ftp localhost
Connected to localhost (127.0.0.1).
220 Unauthorized access, or attempts to gain unauthorized access to this site is prohibited and trespassers will be prosecuted.
Name (localhost:root): anonymous
331 Please specify the password.
Password:
530 Login incorrect.
Login failed.
ftp>
```

Verify that rickywald can login because the user rickywald is listed in /etc/vsftpd.user_list (which lists allowed users since “userlist_deny=NO” in vsftpd.conf) but not in /etc/vsftpd.ftpusers (which lists denied users).

```
[root@GIAC-Linux etc]# ftp localhost
Connected to localhost (127.0.0.1).
220 Unauthorized access, or attempts to gain unauthorized to this site is prohibited and trespassers will be prosecuted.
Name (localhost:root): rickywald
331 Please specify the password.
Password:
230 Login successful. Have fun.
Remote system type is UNIX.
Using binary mode to transfer files.
Ftp>
```

As a validation that even if an entry is in /etc/vsftpd.user_list, if the entry is also in /etc/vsftpd.ftpusers then the end result is that access is denied. For the purpose of this validation, ensure that root is listed in /etc/vsftpd.user_list but also in /etc/vsftpd.ftpusers, then fail to login. After this test, remove root from /etc/vsftpd.user_list.

[root@GIAC-Linux etc]# ftp localhost
Connected to localhost (127.0.0.1).
220 Unauthorized access, or attempts to gain unauthorized to this site is prohibited and trespassers will be prosecuted.
Name (localhost:root): root
331 Please specify the password.
Password:
530 Login incorrect.
Login failed.
Ftp>

Checking file permissions on /var/ftp.

Check that /var/ftp has a uid and gid or 0 and that group and other do not have write access.

[root@GIAC-Linux var]# ls –n /var/ftp
drwxr-xr-x

If we needed to change the permissions, we would have used the “chmod” command.

Secure FTP

Note that if it were not a business requirement for FTP to also be available, then for security reasons we would only enable Secure FTP.

Now verify that you can use secure FTP to connect

su to the user rickywald, because this is the user for which we generated the public/private key pair for ssh connections.
[rickywald@GIAC-Linux ssh]# su rickywald

[rickywald@GIAC-Linux ssh]# sftp localhost
Connecting to localhost...
Enter passphrase for key ‘/home/rickywald/.ssh/id_rsa’:

sftp>

Note that in the above connection no banner is displayed. This will be introduced in the securing openssl section below.

Securing OpenSSL

First use vi to create and then edit the following file:
[root@GIAC-Linux ssh]# vi /etc/ssh/sshd_banner
In this file type a banner message: “Unauthorized access, or attempts to gain unauthorized access to this site is prohibited and trespassers will be prosecuted.”

Use chmod to set the permissions on this file to 400.

This banner will be referenced with the “Banner” option in the /etc/ssh/sshd_config file below.

OpenSSL Configuration file

The OpenSSL configuration file is /etc/ssh/sshd_config. The corresponding man page can be found by typing “man sshd_config”.

Open the configuration file /etc/ssh/sshd_config with vi.
[root@GIAC-Linux ssh]# vi etc/ssh/sshd_config

Uncomment the “Protocol” line and edit this so that it reads:
Protocol 2.
I.e. we are removing protocol 1 because version 1.33 and 1.5 of the SSH protocol are not completely cryptographically safe (found from Nessus).

Change the listen address to the localhost as follows:
ListenAddress 127.0.0.1:22 # ip address of listening port.

Include the path to the banner file that we created:
Banner /etc/ssh/sshd_banner

“DenyGroups” could be used to deny access based on group membership.

Set the following logging level:
LogLevel INFO
As you will find in the man page, the available logging levels are:
QUIET, FATAL, ERROR, INFO, VERBOSE, DEBUG, DEBUG1, DEBUG2 and DEBUG3.

Set “SyslogFacility AUTH”

Set “PermitRootLogin” to “no” so that you cannot directly ssh to Linux-GIAC as root.
Set “PermitEmptyPasswords” to “no” so that empty passwords cannot be used.

Uncomment the lines and modify as necessary so that they read:
RSAAuthentication no #note that this only applies to protocol version 1.
PubkeyAuthentication yes

PasswordAuthenticaton no

Notice that password authentication (which was used earlier to verify that ssh was functioning) is no longer allowed.

Restart sshd as follows, so that the configuration changes are applied.

```
[root@GIAC-Linux /]# service sshd restart
Stopping sshd:                             [   OK   ]
Starting sshd:                             [   OK   ]
```

Demonstrating that you can no longer ssh using password

First fail to ssh to localhost as user rickywald

```
[rickywald@GIAC-Linux /]$ ssh localhost
Permission denied (publickey,keyboard-interactive)
```

Generating user's PKI key pair

Now we will generate a private/public key pair to be used by rickywald. This should be performed on the client machine that rickywald will use to access the server GIAC-Linux

```
[rickywald@client ssl]$ ssh-keygen -b 2048 -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/rickywald/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/rickywald/.ssh/id_rsa.
Your public key has been saved in /home/rickywald/.ssh/id_rsa.pub.
The key fingerprint is:
f5:63:59:e3:fb:c2:55:94:6e:70:0c:b7:01:c1:58:a7 rickywald@GIAC-Linux
[rickywald@client ssl]$
```

In the above, notice that 2048 is the key size in bits and rsa stands for the type of key being created.

Making the server trust the user's Certificate credentials

```
[rickywald@client root]$ cd /home/rickywald/.ssh
[rickywald@client .ssh]$ ls
id_rsa  id_rsa.pub
```

Notice that the key pair for root was placed in /home/rickywald/.ssh

For the following steps, make sure you are logged in as root.

Next, copy the public key /rickywald/.ssh/id_rsa.pub on the client machine to /rickywald/.ssh/authorized_keys on GIAC-Linux, so that GIAC-Linux will trust rickywald's certificate.

[root@GIAC-Linux .ssh]$ cat ./id_rsa.pub >>
/home/rickywald/.ssh/authorized_keys

Change the permissions of the authorized_keys file to 600:
[root@GIAC-Linux ssh] chmod 600 /home/rickywald/.ssh/authorized_keys

Verify that the user can connect over ssh using his public/private key pair

Next we will connect from client, (i.e. The client machine) to GIAC-Linux using ssh and hence using rickywald's public/private key pair.

[rickywald@client /]$ ssh rickywald@GIAC-Linux
<ssh banner message will be displayed here>

Type the passphrase that you set up when root's private/public key was generated.

rickywald@client's passphrase:
<motd banner message will be displayed here>
[rickywald@GIAC-Linux rickywald]$

Even though rickywald is sitting at his client machine, he is connected to the server GIAC-Linux using ssh.

Verify logging

To verify that logging is enabled, ssh to localhost
[rickywald@GIAC-Linux ssh]# ssh localhost
<ouput omitted>

su to root:
[rickywald@GIAC-Linux /]# su root
Password:

Then use the "less" command to view the log file

[root@GIAC-Linux log]# less /var/log/messages

To find the following logging information:

Nov 2 16:03:30 GIAC-Linux sshd[935]: Accepted publickey for rickywald from 127.0.0.1 port 32855 ssh2

IP Kernel Tuning

Use vi to open /etc/sysctl.conf
[root@GIAC-Linux root]# vi /etc/sysctl.conf

As instructed in section 4.1/4.2 of the LinuxBenchmark.pdf and http://www.cymru.com/Documents/ip-stack-tuning.html, add the following lines:

net.ipv4.tcp_max_syn_backlog = 4096
net.ipv4.conf.all.rp_filter = 1
net.ipv4.conf.all.accept_source_route = 0
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.all.secure_redirects = 0

net.ipv4.conf.default.accept_source_route = 0
net.ipv4.conf.default.accept_redirects = 0
net.ipv4.conf.default.secure_redirects = 0

Then add the following lines so that Linux-GIAC does not redirect network traffic.
net.ipv4.conf.all.send_redirects = 0
net.ipv4.conf.default.send_redirects = 0

Verifying that failed logins are logged

If you are already logged in, log out of GIAC-Linux. Take a note of the time so that we can compare this with the time written in the log file, (which will become clearer in a few moments). Attempt to login with the username rickywald, but purposely type an incorrect password. Then login as root.

View the messages log file as follows:
[root@GIAC-Linux log]#less /var/log/messages
<output omitted>
Nov 12 18:02:15 GIAC Linux gdm(pam_unix)[737]: authentication failure; logname= uid=0 euid=0 tty=:0 ruser=gdm rhost=localhost
 user=rickywald
<output omitted>

Thus the failed login has been correctly recorded.

Verify that successful logins are recorded.

If you are already logged in, log out of GIAC-Linux. Take a note of the time so that we can compare this with the time written in the log file, (which will become clearer in a few moments). Login as rickywald.

View the messages log file as follows:
(note that you will need to su to root first).
[root@GIAC-Linux log]#less /var/log/messages
<output omitted>
Nov 12 18:12:08 GIAC-Linux gdm(pam_unix)[737]: session opened for user rickywald by (uid=0)
<output omitted>

Thus the successful login has been correctly recorded.

Now repeat this with the root account.

Preventing devices from being mounted on partitions.

We are now going to edit /etc/fstab so that certain partitions are mounted such that devices cannot be wrongly mounted on these partitions. This is performed by adding the “nodev” option. We will do this to /usr, /usr/local, /var, /tmp, /home, /mnt/cdrom.

On /mnt/cdrom we will also use the nosuid option to prevent software on the CDROM from running on GIAC-Linux with the set uid bit set.

[root@GIAC-Linux etc]# vi /etc/fstab

Once the changes are made, view the /etc/fstab file to verify that the changes have indeed been made:

```
[root@GIAC-Linux etc]# more /etc/fstab
LABEL=/             /                   ext3    defaults        1 1
none                /dev/pts            devpts  gid=5,mode=620  0 0
LABEL=/home         /home               ext3    defaults,nodev  1 2
none                /proc               proc    defaults        0 0
none                /dev/shm            tmpfs   defaults        0 0
LABEL=/tmp          /tmp                ext3    defaults,nodev  1 2
LABEL=/usr          /usr                ext3    defaults,nodev  1 2
LABEL=/usr/local    /usr/local          ext3    defaults,nodev  1 2
```

```
LABEL=/var          /var                ext3    defaults,nodev      1 2
/dev/hda7           swap                swap    defaults      0 0
/dev/cdrom          /mnt/cdrom          udf,iso9660
noauto,owner,kudzu,ro,nodev,nosuid 0 0
```

After rebooting GIAC-Linux, issue the mount command or view the /etc/mtab file to verify that these changes have been applied.

```
[root@GIAC-Linux root]# mount
/dev/hda1 on / type ext3 (rw)
none on /proc type proc (rw)
usbdevfs on /proc/bus/usb type usbdevfs (rw)
none on /dev/pts type devpts (rw,gid=5,mode=620)
/dev/hda6 on /home type ext3 (rw,nodev)
none on /dev/shm type tmpfs (rw)
/dev/hda3 on /tmp type ext3 (rw,nodev)
/dev/hda2 on /usr type ext3 (rw,nodev)
/dev/hda5 on /usr/local type ext3 (rw,nodev)
/dev/hda8 on /var type ext3 (rw,nodev)
```

```
[root@GIAC-Linux root]# more /etc/mtab
/dev/hda1 / ext3 rw 0 0
none /proc proc rw 0 0
usbdevfs /proc/bus/usb usbdevfs rw 0 0
none /dev/pts devpts rw,gid=5,mode=620 0 0
/dev/hda6 /home ext3 rw,nodev 0 0
none /dev/shm tmpfs rw 0 0
/dev/hda3 /tmp ext3 rw,nodev 0 0
/dev/hda2 /usr ext3 rw,nodev 0 0
/dev/hda5 /usr/local ext3 rw,nodev 0 0
/dev/hda8 /var ext3 rw,nodev 0 0
```

Remove privileges on accessing removable media

Use vi to open the console.perms file as follows:
```
[root@GIAC-Linux security]# vi /etc/security/console.perms
```

We are about to remove the privileges that allow normal users to access removable media.

In the Permission Definitions section, comment out lines so that you obtain the following:

```
# permission definitions
```

```
#<console>  0660 <floppy>     0660 root.floppy
<console>  0600 <sound>      0600 root
#<console>  0600 <cdrom>      0660 root.disk
#<console>  0600 <pilot>      0660 root.uucp
#<console>  0600 <jaz>        0660 root.disk
#<console>  0600 <zip>        0660 root.disk
#<console>  0600 <ls120>      0660 root.disk
<console>  0600 <scanner>    0600 root
#<console>  0600 <camera>     0600 root
#<console>  0600 <memstick>   0600 root
#<console>  0600 <flash>      0600 root
#<console>  0600 <diskonkey>  0660 root.disk
#<console>  0600 <rem_ide>    0660 root.disk
<console>  0600 <fb>         0600 root
<console>  0600 <kbd>        0600 root
<console>  0600 <joystick>   0600 root
<console>  0600 <v4l>        0600 root
<console>  0700 <gpm>        0700 root
<console>  0600 <mainboard>  0600 root
#<console>  0600 <rio500>     0600 root
```

Create honeypot accounts

Use the adduser command to create user accounts that will not actually be given out to users. Then if the logs show that someone is attempting to or successfully logging in with one of these accounts as described in Verifying that failed logins are logged, you know that you are being attacked. You may wish to set the default shell to /dev/null.

```
[root@GIAC-Linux root]: adduser johndoe –s /dev/null
[root@GIAC-Linux root]: passwd johndoe
Changing password for user Johndoe3
New password:
Retype new password:
Passwd: all authentication tokens updated successfully.
```

Stop listening on port 6000.

Issue the netstat command to see that GIAC-Linux is listening on port 6000.

```
[root@GIAC-Linux xdm]# netstat -an
<output omitted>
tcp        0      0 0.0.0.0:6000            0.0.0.0:*               LISTEN
<output omitted>
```

We are going to edit the Xservers file so that X servers do not listen on TCP port 6000.

[root@GIAC-Linux xdm]# vi /etc/X11/gdm/gdm.conf

Add “–nolisten tcp” to the Standard server sections, so that the following is displayed:
“[server-Standard]
name=Standard server
command=/usr/X11R6/bin/X –nolisten tcp”

When the machine is rebooted, issue the “netstat –an” command again to verify that GIAC-Linux is no longer listening on port 6000.

Restrict access to “at” and “cron”

Recall, we have already learnt about “cron” and “at” as follows:

Atd – This runs jobs queued by at. “At” runs jobs from the standard input, i.e. the keyboard, when those jobs are supposed to run at a later time.

Crond – This runs in the background to execute scheduled commands.

We want to limit the issuing of the “at” and “cron” commands to root. /etc/cron.allow and /etc/at.allow list the users that can access the “cron” and “at” commands respectively. Thus we will make sure that the only entry in these two files is root.

at

Create the file /etc/at.allow. Once this file is created, only accounts listed in this file can access the "at" service.
[root@GIAC-Linux etc]# touch /etc/at.allow
[root@GIAC-Linux etc]# echo root >> /etc/at.allow

Next create a test file.

[root@GIAC-Linux etc]# vi /etc/test_file
Add the following text: “echo The /etc/test_file has run!”
Change the permissions so that nobody can modify this file and so that only root can execute this file.
[root@GIAC-Linux etc]# chmod 500 /etc/test_file
Run this test file.
[root@GIAC-Linux etc]# /etc/test_file
The /etc/test_file has run!

Issue the "at" command to verify that root can issue the "at" command.

```
[root@GIAC-Linux etc]# at -f /etc/test_file 11:42
warning: commands will be executed using (in order) a) $SHELL b) login shell c)
/bin/sh
job 1 at 2003-11-05 11:42
```

Next, su to the user rickywald and verify that you are unable to issue the "at" command.

```
[root@GIAC-Linux etc]# su rickywald
[rickywald@GIAC-Linux etc]$ at -f /etc/test_file 11:43
You do not have permission to use at.
```

Cron

Create the file /etc/cron.allow.

```
[root@GIAC-Linux etc]# touch /etc/cron.allow
```

Add the root user to the list of allowed users.

```
[root@GIAC-Linux etc]# echo root >> /etc/cron.allow
```

Verify the text within the /etc/cron.allow file.

```
[root@GIAC-Linux etc]# more /etc/cron.allow
root
```

Next, successfully issue the crontab command as user root but unsuccessfully as user rickywald.

```
[root@GIAC-Linux etc]# crontab -u root -l
no crontab for root
[root@GIAC-Linux etc]# crontab -u rickywald -l
You (rickywald) are not allowed to use this program (crontab)
See crontab(1) for more information
[root@GIAC-Linux etc]#
```

Note that -u is used to specify the user, whilst -l lists the crontab for that user.

Next, make sure that the only access given to the cron and at allow files are read access to root.

```
[root@GIAC-Linux etc]# chmod 400 /etc/cron.allow /etc/at.allow
[root@GIAC-Linux etc]# ls -n /etc/cron.allow /etc/at.allow
-r--------    1 0        0              58 Nov  5 11:32 /etc/at.allow
-r--------    1 0        0              53 Nov  5 11:46 /etc/cron.allow
[root@GIAC-Linux etc]#
```

Next we will set the appropriate permissions on /etc/crontab, /var/spool/cron and the various /etc/cron.* files. We want to restrict normal user access to these files because cron runs as root and crontab has the setuid bit set.

/etc/crontab
```
[root@GIAC-Linux root]# ls -n /etc/crontab
-rw-r--r--    1 0        0             255 Feb  7  2003 /etc/crontab
```
Notice that /etc/crontab is owned and has a group of root, which is fine. However the only access required is for root to have read access.
```
[root@GIAC-Linux root]# chmod 400 /etc/crontab
[root@GIAC-Linux root]# ls -n /etc/crontab
-r--------    1 0        0             255 Feb  7  2003 /etc/crontab
[root@GIAC-Linux root]#

[root@GIAC-Linux etc]# ls -n /etc/cron.allow /etc/cron.d /etc/cron.daily
/etc/cron.hourly /etc/cron.monthly /etc/crontab /etc/cron.weekly
-r--------    1 0        0              53 Nov  5 11:46 /etc/cron.allow
-r--------    1 0        0             255 Feb  7  2003 /etc/crontab

/etc/cron.d:
total 0

/etc/cron.daily:
total 24
lrwxrwxrwx    1 0        0              28 Oct 20 20:19 00-logwatch ->
../log.d/scripts/logwatch.pl
-rwxr-xr-x    1 0        0             276 Jan 24  2003 0anacron
-rwxr-xr-x    1 0        0              51 Jan 24  2003 logrotate
-rwxr-xr-x    1 0        0             418 Feb 10  2003 makewhatis.cron
-rwxr-xr-x    1 0        0             104 Feb 27  2003 rpm
-rwxr-xr-x    1 0        0             132 Feb 19  2003 slocate.cron
-rwxr-xr-x    1 0        0             193 Feb 10  2003 tmpwatch

/etc/cron.hourly:
total 0

/etc/cron.monthly:
total 4
-rwxr-xr-x    1 0        0             278 Jan 24  2003 0anacron

/etc/cron.weekly:
total 8
-rwxr-xr-x    1 0        0             277 Jan 24  2003 0anacron
-rwxr-xr-x    1 0        0             414 Feb 10  2003 makewhatis.cron
[root@GIAC-Linux etc]#
```

All of these correctly have a uid and gid of 0.

Now from group and other we remove all rights:
[root@GIAC-Linux etc]# chmod -R go-rwx /etc/cron.*

Next verify that the change has been made.

[root@GIAC-Linux etc]# ls -n /etc/cron.allow /etc/cron.d /etc/cron.daily /etc/cron.hourly /etc/cron.monthly /etc/crontab /etc/cron.weekly
(for shorthand we could have typed “ls –n /etc/cron.*”)

```
-r--------    1 0       0            53 Nov  5 11:46 /etc/cron.allow
-r--------    1 0       0           255 Feb  7  2003 /etc/crontab

/etc/cron.d:
total 0

/etc/cron.daily:
total 24
lrwxrwxrwx    1 0       0            28 Oct 20 20:19 00-logwatch ->
../log.d/scripts/logwatch.pl
-rwx------    1 0       0           276 Jan 24  2003 0anacron
-rwx------    1 0       0            51 Jan 24  2003 logrotate
-rwx------    1 0       0           418 Feb 10  2003 makewhatis.cron
-rwx------    1 0       0           104 Feb 27  2003 rpm
-rwx------    1 0       0           132 Feb 19  2003 slocate.cron
-rwx------    1 0       0           193 Feb 10  2003 tmpwatch

/etc/cron.hourly:
total 0

/etc/cron.monthly:
total 4
-rwx------    1 0       0           278 Jan 24  2003 0anacron

/etc/cron.weekly:
total 8
-rwx------    1 0       0           277 Jan 24  2003 0anacron
-rwx------    1 0       0           414 Feb 10  2003 makewhatis.cron
[root@GIAC-Linux etc]#
```

Setting MySQL root password

We are now going to set a MySQL root password.

```
[root@GIAC-Linux bin]# /usr/bin/mysqladmin -u root  password 'new-password'
[root@GIAC-Linux bin]# /usr/bin/mysqladmin -u root -h GIAC-Linux  password
'new-password'
```

Next we verify that we can no longer access the mysql database using the mysqlshow command as we were previously able to.

```
[root@GIAC-Linux etc]#mysqlshow
mysqlshow: Access denied for user: 'root@localhost'  (Using password:  NO)
```

Next we use the –p tag with the "mysqlshow" command so that we can present the mysql password and hence access the database.

```
[root@GIAC-Linux etc]#mysqlshow –p
Enter password:
+-----------+
| Databases |
+-----------+
| mysql     |
| test      |
+-----------+
```

Access control to xinetd

Implementing simple access controls in xinetd in addition to the TCP wrappers that we setup.

To /etc/xinetd.conf, add the following line "only_from = X.X.X.X/Y", where x.x.x.x is the network address of the network that will be accessing services controlled by xinetd, which in our case is vsftpd. Y is the subnet mask for this network, for example 24 for a class C network.

To verify this, fail to "ftp localhost" from a system outside of the x.x.x.x/y network and successfully "ftp localhost" from a system inside the x.x.x.x/y network.

After making the changes, remember to restart xinetd with the "service xinetd restart" command.

Preventing root login other than from the console.

If there are multiple people with the root password and illegal action is recorded, how do we know who caused the action. For these and other reasons it is beneficial to not let root directly log in to the system. Instead the user can change to root once they have logged in. Note that direct root access when at the console will be left on in case of emergency.

To prevent direct root login:
Open the following file: [root@GIAC-Linux etc]# vi /etc/securetty

Delete “ttyn” and “vc/n” for 2<= n <=11.
Save the changes.

The man pages describe securetty as “ file contains the device names of tty lines on which root is allowed to login.”

Set null as default shell for system accounts

For the system accounts we should change the default shell to /dev/null as this is not a proper shell and hence reduces the impact of an attacker gaining access to one of these accounts. This can be achieved by modifying the /etc/passwd file.

Here is an example from the /etc/passwd file:
root:x:0:0:root:/root:/bin/bash
root is the account name, x is a replacement for the password which is stored in /etc/shadow in an encrypted form, 0 is the User ID, 0 is the Group ID, root is a friendly name for the user, /root is the accounts home folder, /bin/bash is the default shell.

For the following line:
bin:x:1:1:bin:/bin:/sbin/nologin
We will change this to:
bin:x:1:1:bin:/bin:/dev/null
Repeat this for all the system accounts, rather than real user accounts.

Snort

Installing libpcap

Libpcap is a prerequisite for snort.

Go to http://www.tcpdump.org/ and download libpcap-0.8.1-316.tar.gz.

Gunzip then extract the files from the archive.
[root@GIAC-Linux snort]# gunzip /securitytools/snort/libpcap-0.8.1-316.tar.gz
[root@GIAC-Linux snort]# tar xf /securitytools/snort/libpcap-0.8.1-316.tar

The installation instructions accompanying the software can be viewed by typing:
[root@GIAC-Linux libpcap-0.8.1]# more /securitytools/snort/libpcap-0.8.1/INSTALL.txt

Enter these commands to install libpcap.
[root@GIAC-Linux libpcap-0.8.1]# ./configure
[root@GIAC-Linux libpcap-0.8.1]# make

[root@GIAC-Linux libpcap-0.8.1]# make install

Installing Snort.

Download snort (snort-2.0.2.tar.gz) from http://www.snort.org/dl/

Next, check the md5 hash values.
[root@GIAC-Linux snort]# more ./snort-2.0.2.tar.gz.md5
md5 : 21b14d90e2a323831d85f3d845d64b23 snort-2.0.2.tar.gz
sha1 : e79733bc1a17b2cb9cb0d64cc94c7cc9d16fba18 snort-2.0.2.tar.gz
[root@GIAC-Linux snort]# md5sum snort-2.0.2.tar.gz
21b14d90e2a323831d85f3d845d64b23 snort-2.0.2.tar.gz
[root@GIAC-Linux snort]#

Gunzip then extract the files from the archive.
[root@GIAC-Linux snort]# gunzip /securitytools/snort/snort-2.0.2.tar.gz
[root@GIAC-Linux snort]# tar xf /securitytools/snort/snort-2.0.2.tar

The installation instructions accompanying the software can be viewed by typing:
[root@GIAC-Linux doc]# more /securitytools/snort/snort-2.0.2/doc/INSTALL
Enter these commands to install snort.
[root@GIAC-Linux snort-2.0.2]# /securitytools/snort/snort-2.0.2/configure
[root@GIAC-Linux snort-2.0.2]# make
[root@GIAC-Linux snort-2.0.2]# make install

Verify that snort can capture packets.

To verify that snort will run and can see the packets on the Ethernet port, start snort running in verbose mode
[root@GIAC-Linux snort-2.0.2]# snort -v

This command will show the TCP/IP headers, for example:

=+=
s11/04-11:58:42.429335 x.x.x.x -> x.0.0.x
EIGRP TTL:2 TOS:0xC0 ID:0 IpLen:20 DgmLen:60
=+=
<IP address changed to x.x.x.x for confidentiality>

Press ctrl+c to stop snort verbose mode.

Configuring snort (logging and listening on Ethernet port)

First create the directory that will hold the snort logs.
[root@GIAC-Linux log]# mkdir /var/log/snort
[root@GIAC-Linux log]# chmod 700 /var/log/snort

Note that due to the installation directory, snort.conf is located at /securitytools/snort/snort-2.0.2/etc/snort.conf

Using vi, edit snort.conf by un-commenting “var HOME_NET $eth0_ADDRESS” so that the Ethernet port will be used by snort.

The following command will start snort using the default log location /var/log/snort and will use the rules configuration file snort.conf.

[root@GIAC-Linux etc]# cd /securitytools/snort/snort-2.0.2/etc/
[root@GIAC-Linux etc]# snort -D -c snort.conf

Next verify that snort is running properly by checking the log files as follows.

[root@GIAC-Linux snort]# more /var/log/snort/alert

Password Policy

We are going to use the variables in /etc/login.defs to set the maximum value for the number of days between a password change (PASS_MAX_DAYS), the minimum number of days in which it is not allowed to change a password (PASS_MIN_DAYS), the minimum password length (PASS_MIN_LEN) and the minimum length of time between a warning message and forcing the user to change the password (PASS_WARN_AGE).

Use vi to edit the file.
[root@GIAC-Linux etc]# vi /etc/login.defs

Change the following variables to take these values.
PASS_MAX_DAYS 30
PASS_MIN_DAYS 7
PASS_MIN_LEN 8
PASS_WARN_AGE 7

We will now apply these values for PASS_MIN_DAYS, PASS_MIN_LEN and PASS_WARN_AGE, but applying these values directly to “rickywald”.
Issue the following commands:
[root@GIAC-Linux etc]# chage -m 7 -M 30 -W 7 rickywald
Repeat this for the other user accounts.

-m, -M and -W corresponds to PASS_MIN_DAYS, PASS_MAX_DAYS and PASS_WARN_AGE respectively.

As a quick test, fail to change the password for rickywald to a value shorter than 7 characters.

```
[rickywald@GIAC-Linux etc]$ passwd
Changing password for user rickywald.
Changing password for rickywald
(current) UNIX password:
New password:
BAD PASSWORD: it's WAY too short
New password:
BAD PASSWORD: it's WAY too short
New password:
BAD PASSWORD: it's WAY too short
passwd: Authentication token manipulation error
[rickywald@GIAC-Linux etc]$
```

Look in the file /etc/shadow and notice the corresponding entries for rickywald:

```
[root@GIAC-Linux etc]# more /etc/shadow
rickywald:$1$TbcOwCfV$shbBor1GX9dCUPqyfqVRQ/:12364:5:30:7:::
```

IP Tables

IP Tables is effectively a host based firewall for Linux. From the man pages we now highlight some of the tags/options that we will use.

--append	This adds rules to the end of a selected chain.
Filter	This is the default table (i.e. this is used if –t is not specified).
INPUT	Refers to packets entering into the system.
OUTPUT	Refers to locally-generated packets.
ACCEPT	A packet that matches the corresponding rule can be accepted, so that the packet is allowed.
DROP	A packet that matches the corresponding rule will be dropped.
--list	Lists all the rules in the selected chain.
--protocol	The protocol of the rule or packet to check.
--source	IP address or hostname or the originator of the packet.
--destination	IP address or hostname of the destination of the packet.
--source-port	Source port
--destination-port	Destination port

Rules can be added as in the following example:

```
[root@GIAC-Linux rickywald]# iptables --append INPUT --protocol tcp --
destination 192.168.0.2 --destination-port 80 --jump ACCEPT
[root@GIAC-Linux rickywald]# iptables --append INPUT --protocol tcp --source
192.168.1.0/24 --destination 192.168.0.2 --destination-port 21 --jump ACCEPT
[root@GIAC-Linux rickywald]# iptables --append INPUT --protocol tcp --source
192.168.3.2 --destination 192.168.0.2 --destination-port 22 --jump ACCEPT
```

The rules can be created or modified to fit the specific situation/environment. For incoming rules, the default is to drop anything not specified by the created rules.

Use the iptables command with the --list tag to view the rules that we have just created.

```
[root@GIAC-Linux rickywald]# iptables --list
Chain INPUT (policy ACCEPT)
Target     prot opt source          destination
ACCEPT     tcp  --  anywhere        192.168.0.2  tcp dpt:ftp
<output omitted>
```

Note that the border routers in this example network are configured to not allow incoming packets with a private source address, which does provide some protection from the corresponding type of external attacks.

Use the following command to save the iptables rule set that we have just created.

```
[root@GIAC-Linux rickywald]# service iptables save
Saving current rules to /etc/sysconfig/iptables:          [ OK ]
```

Check default permissions.

As an example, check what the default permissions are for newly created files in the following situations.

```
[root@GIAC-Linux etc]# su rickywald
[rickywald@GIAC-Linux home]$ cd /home/rickywald
[rickywald@GIAC-Linux rickywald]$ touch test_file2
[rickywald@GIAC-Linux rickywald]$ ls -n test_file2
-rw-rw-r--  1 500  500  0 Nov  8 12:08 test_file2
```

Now we are going to use vi to open up each of the following files and set the umask value to 077. If umask is not already created then create it, or if it is created, change the value to 077.

```
[root@GIAC-Linux etc]# vi /etc/profile
set umask to 077
```

```
[root@GIAC-Linux etc]# vi /etc/csh.login
set umask to 077
[root@GIAC-Linux etc]# vi /etc/csh.cshrc
set umask to 077
[root@GIAC-Linux etc]# vi /etc/bashrc
set umask to 077
[root@GIAC-Linux etc]# cd /root
[root@GIAC-Linux root]# vi /root/.bash_profile
set umask to 077
[root@GIAC-Linux root]# vi /root/.bashrc
set umask to 077
[root@GIAC-Linux root]# vi /root/.cshrc
set umask to 077
[root@GIAC-Linux root]# vi /root/.tcshrc
set umask to 077
```

Now we repeat the above test of creating a new file as the user rickywald and checking what the default permissions are.

```
[rickywald@GIAC-Linux rickywald]$ touch test_file4
[rickywald@GIAC-Linux rickywald]$ ls -n test_file4
-rw-------    1 500      500             0 Nov  8 12:23 test_file4
```

You can also use the “umask” command to determine the umask

```
[rickywald@GIAC-Linux rickywald]$ umask
0077
```

Verify permissions and ownership

Next verify the permissions and ownership, checking that all files have an owner and group of root. Furthermore the /etc files should only have octet permissions equal to 444 (i.e. read access to everyone).

```
[root@GIAC-Linux root]# ls -an /etc/profile /etc/csh.login /etc/csh.cshrc
/etc/bashrc /root/.bash_profile /root/.bashrc /root/.cshrc /root/.tcshrc
-rw-r--r--    1 0        0            1497 Nov  8 12:20 /etc/bashrc
-rw-r--r--    1 0        0             561 Nov  8 12:19 /etc/csh.cshrc
-rw-r--r--    1 0        0             420 Nov  8 12:19 /etc/csh.login
-rw-r--r--    1 0        0             853 Nov  8 12:18 /etc/profile
-rw-r--r--    1 0        0             244 Nov  8 12:21 /root/.bash_profile
-rw-r--r--    1 0        0             187 Nov  8 12:21 /root/.bashrc
-rw-r--r--    1 0        0             221 Nov  8 12:22 /root/.cshrc
-rw-r--r--    1 0        0             207 Nov  8 12:22 /root/.tcshrc
```

Use the chmod command to change the permissions of the /etc files to 444 as follows.

[root@GIAC-Linux root]# chmod 444 /etc/profile /etc/csh.login /etc/csh.cshrc /etc/bashrc

Verify the permissions have been correctly modified.

```
[root@GIAC-Linux root]# ls -an /etc/profile /etc/csh.login /etc/csh.cshrc
/etc/bashrc /root/.bash_profile /root/.bashrc /root/.cshrc /root/.tcshrc
-r--r--r--  1 0    0        1497 Nov  8 12:20 /etc/bashrc
-r--r--r--  1 0    0         561 Nov  8 12:19 /etc/csh.cshrc
-r--r--r--  1 0    0         420 Nov  8 12:19 /etc/csh.login
-r--r--r--  1 0    0         853 Nov  8 12:18 /etc/profile
-r--r--r--  1 0    0         244 Nov  8 12:21 /root/.bash_profile
-r--r--r--  1 0    0         187 Nov  8 12:21 /root/.bashrc
-r--r--r--  1 0    0         221 Nov  8 12:22 /root/.cshrc
-r--r--r--  1 0    0         207 Nov  8 12:22 /root/.tcshrc
```

Find world write access files

Issue the following commands to find files that can be written to by anyone.
[root@GIAC-Linux /]# find / -depth –type f – perm –o+w –fprint /root/list_write_oth
[root@GIAC-Linux /]# more /root/list_write_file

Issue the following commands to find files that can be written to by members of the group.
[root@GIAC-Linux /]# find / -depth –type f – perm –g+w –fprint /root/list_write_gr
[root@GIAC-Linux /]# more /root/list_write_gr

The chmod command can be used to change permissions as appropriate.

Preventing core dumps

We are next going to add the following text to /etc/security/limits.conf
“* soft core 0
* hard core 0”
What this will do is for everyone (indicated by the wildcard *), only allow core dumps to have a maximum size of 0 (KB), which effectively means that core dumps are not allowed for any user. The reason this change is made is because a hacker could gain valuable information from viewing the information in a core dump.

Use vi to make this change.

Preventing su

We are going to limit “su” to a specific group, so that only members of this group can su. This means that if an attacker is able to compromise a normal account, there is a lower risk of the attacker using su to escalate their privileges to root.

Change the group of /bin/su to wheel
[root@GIAC-Linux root]# chgrp wheel /bin/su

Change the permissions so that only root and members of wheel can execute the su command.
[root@GIAC-Linux root]# chmod o-rwx /bin/su

Use the “usermod” command on the user rickywald to change rickywald so that he belongs to the wheel group.
[root@GIAC-Linux root]# usermod –G wheel rickywald

To verify that this is working as expected:
- Login as rickywald and successfully su to root.
    [rickywald@GIAC-Linux rickywald]$ su root
    Password:
    [root@GIAC-Linux rickywald]$
- Login as another user and fail to su.
    [user3@GIAC-Linux user3]$ su root
    bash: /bin/su: Permission denied.

Repeat the verification that Apache, MySQL and VSFTP servers are run at startup

Now that the hardening has been performed, as availability is part of security, follow the steps taken in Verify that Apache, MySQL and VSFTP servers are run at startup (except using the –p tag after mysqlshow as described in Setting MySQL root password) to confirm that Apache, MySQL and VSFTP are still running at startup.

Note that, to perform this test on vsftpd, we will need (at least temporarily) to include 127.0.0.1 in the allowed IP addresses that can connect to xinetd services, (i.e. vsftpd in this example). This is set in /etc/xinetd.conf as described in Access control to xinetd.

Section 8 – Ongoing Maintenance

In this section we shall list the actions to be performed on an ongoing basis.

CIS Scan

As we did in section 5, use the /usr/local/CIS/cis-scan command to carry out a cis-scan. The score at the end of section 7 was 8.15, which is considerably higher than before any hardening was performed, where the base line value was previously 4.92. Then use the command:
egrep "^Negative" /usr/local/CIS/cis-ruler-log.20031109-16:52:26.1046 (with the appropriate date replaced)
to view the points which lead to a non-perfect score. You will notice that some of the outstandings are business requirements, whilst others are false positives.

With our new baseline, perform this CIS Scan on a periodic basis as part of ongoing maintenance.

Nessus

As we did in the Running Nessus section, use "/usr/local/sbin/nessusd –D" to start the Nessus daemon, then use "/usr/local/bin/nessus" to start to nessus client. Then login and start the scan. Since hardening you will notice a substantially improved vulnerability assessment results.

Periodically on an ongoing basis perform a Nessus scan with the latest nessus software to detect new vulnerabilities and validate that no change to the system has opened up old vulnerabilities. Take the appropriate action to resolution.

Training

Users/Staff should be trained on security, which should include social engineering and physical security. Training should be ongoing and should include refresher courses.

Patches & Upgrades

As we did in the Retrieving and installing packages with up2date section, use the up2date agent to install the latest patches.

Also ensure that you upgrade separately installed software to recent versions which include fixes to security flaws.

Verify running processes

Use the "ps –ef" command to view which processes are running and validate that the running processes are as expected. Note that this is not full proof as an attacker could attempt to modify the ps command and hence hide their tracks.

```
[root@GIAC-Linux root]# ps –ef
UID    PID   PPID   C   STIME   TTY    TIME       CMD
root     1      0      0    15:56    ?      00:00:04   init
…
<output omitted>
```

Verify listening ports

Use the netstat –an command to view ports GIAC-Linux is listening on.

```
[root@GIAC-Linux root]# netstat –an
<output omitted>
Proto  Recv-Q Send-Q  Local Address   Foreign Address   State
tcp       0          0            0.0.0.0:80       0.0.0.0.*              LISTEN
…
<output omitted>
```

A list and description of port numbers can be found at http://www.iana.org/assignments/port-numbers and below is a small subset.

Port	Service
21	FTP
22	SSH
80	HTTP
3306	MySQL

Backups

Using the datacentres backup facilities, take a weekly full backup and a daily incremental backup. For example the "tar" command can be used to create archives and these archives can be transferred using the "scp" command. Furthermore one copy of these backups should be kept in a secure area onsite (i.e. in the datacentre), whilst another copy should be kept off site in a secure location.

Regularly test restoring the system using backups.

Also you should keep detailed and up to date configuration documentation on the system. This again should be securely stored.

View logs

Regularly view the following logs:

- /var/log/messages
- /var/log/mysqld.log
- /var/log/secure/
- /var/log/vsftpd.log
- /var/log/httpd/access_log
- /var/log/snort/*

We should pay particular attention to the honeypot accounts that we set up in Create honeypot accounts.

Newsgroups

Subscribe to appropriate news groups that will inform you of relevant security information on vulnerabilities and what to do about them. For example you could use "@Risk" from SANS, which is found at http://www.sans.org/newsletters/risk/. This now contains details of what large organizations have done to reduce the impact of these vulnerabilities.

Datacentre

Perform regular audits of the datacentre. Regularly check that the site UPS, power generators and monitoring equipment are functioning to the expected levels.

Password Strengths

We are going to use a password cracking tool to try and crack the passwords in the /etc/shadow file. This will be used to check that strong passwords are being used.

Obtain a copy of John the Ripper 1.6 (Unix – sources, 486 KB) password cracker (john-1.6.tar.gz). In this example we used a trusted version from a CD-ROM, however this software can be downloaded from http://www.openwall.com/john. We placed this into a new directory called /securitytools/cracker. Note that any local directory could be used, but we pick a specific directory to assist in communication.

Check that /securitytools/cracker has a uid and guid of 0 and that only root has permissions.

Next, gunzip then un-tar the file.
[root@GIAC-Linux cracker]# gunzip john-1.6.tar.gz
[root@GIAC-Linux cracker]# tar xf john-1.6.tar

The installation instructions accompanying the source code can be viewed as follows:
[root@GIAC-Linux doc]# more /securitytools/cracker/john-1.6/doc/INSTALL

From the src directory, type “make” to determine the supported operating systems.

[root@GIAC-Linux cracker]# cd /securitytools/cracker/john-1.6/src
[root@GIAC-Linux cracker]# make
<output omitted>
linux-x86-any-elf
<output omitted>

Install John the Ripper using the make command.

[root@GIAC-Linux cracker]# make linux-x86-any-elf

John the ripper can be run with the following command:

[root@GIAC-Linux cracker]# /securitytools/cracker/john-1.6/run/john /etc/shadow
Loaded 4 passwords with 4 different salts <output omitted>
Apples0 (johndoe4)
Guesses:1 time: 0:00:00:29 100% (2) c/s: 2928 trying: Apples0

So this password was cracked extremely quickly. Using this method we, rather than a black hat, can find the weak passwords.

This should be performed periodically on an ongoing basis.

References

Red Hat Linux 9 Man pages.

http://www.acnc.com/04_01_00.html (Author: Advanced Computer & Network Corporation, Title: RAID.edu, Last Updated: 2003, Date Accessed: Nov 2003)

www.openssl.org or more specifically http://www.openssl.org/docs/apps/openssl.html (Authors: OpenSSL members. Members can be found at http://www.openssl.org/about/, Title: openssl(1), Last Updated: Unknown, Date Accessed: 1 Nov 2003)

www.apache.org or more specifically http://httpd.apache.org/docs-2.0/ including http://httpd.apache.org/docs-2.0/install.html, http://httpd.apache.org/docs-2.0/invoking.html, http://httpd.apache.org/docs-2.0/misc/security_tips.html (Authors: Apache HTTP Server documentation project with people including those listed at http://httpd.apache.org/docs-project/, Titles: Compiling and Installing, Starting Apache and Security Tips respectively. Last Updated: Unknown, Date Accessed: Nov 2003)

http://www.mysql.com/documentation/mysql/bychapter/manual_Installing.html#Linux-RPM (Authors: MySQL AB, Title: Installing MySQL, Last Updated: Unknown, Date Accessed: Nov 2003)

http://www.mysql.com/documentation/mysql/bychapter/manual_Installing.html#Verifying_Package_Integrity (Authors: MySQL AB, Title: Installing MySQL, Last Updated: Unknown, Date Accessed: Nov 2003)

http://www.mysql.com/documentation/mysql/bychapter/manual_Installing.html#Post-installation(Authors: MySQL AB, Title: Installing MySQL, Last Updated: Unknown, Date Accessed: Nov 2003)

www.cisecurity.org or more specifically http://www.cisecurity.org/bench_linux.html including LinuxBenchmark.pdf and README found under CIS Security Benchmarks and Scoring Tools for Linux Level 1. (Authors: CIS Security, Title: CIS Security Benchmarks and Scoring Tools for Linux Level 1, Last Updated: Unknown, Date Accessed: Nov 2003)

ftp://vsftpd.beasts.org/users/cevans/untar/vsftpd-1.2.0/EXAMPLE/INTERNET_SITE/README (Authors: Chris Evans, Title: vsftpd-1.2.0/EXAMPLE/INTERNET_SITE/README, Last Updated: Unknown, Date Accessed: Nov 2003)

ftp://vsftpd.beasts.org/users/cevans/untar/vsftpd-1.2.0/EXAMPLE/INTERNET_SITE/vsftpd.xinetd (Authors: Chris Evans, Title:

vsftpd-1.2.0/EXAMPLE/INTERNET_SITE/README, Last Updated: Unknown, Date Accessed: Nov 2003)

http://www.redhat.com/archives/rhl-beta-list/2003-September/msg00099.html, which points to https://rhn.redhat.com/help/RHNS-CA-CERT (Authors: Michael Young, Last Updated: Wed, 3 Sep 2003 09:28:26, Date Accessed: Nov 2003)

http://www.cymru.com/Documents/ip-stack-tuning.html (Authors: Rob Thomas, Last Updated: 3 Dec 2000, Date Accessed: Nov 2000).

http://www.iana.org/assignments/port-numbers (Authors: IANA, Last Updated: 2003-11-18, Date Accessed: Nov 2000).

Note: The companies and people noted in this document other than in the reference section are purely fictional.