Global Information Assurance Certification Paper

GIAC Certified UNIX Security Administrator (GCUX)

Practical Assignment Version 2.0

Option 1 - Securing Unix Step by Step

By Bernd M. Constant

1 Abstract

This paper takes a look at hardening a server that will be set in a LAMP setting (Linux, Apache, MySQL, PHP). This paper guides you from the initial installation though to how to customize configurations to provide the required functionality without sacrificing the need for security.

Index

1 Abstract 2
2 Server Specification and Risk Mitigation Plan 6
2.1 What role does this server fulfill? 6
2.2 What are the hardware requirements? 7
2.3 What is the specific version of the operating system? 7
2.4 What third party software will be used? 7
2.5 What services will be offered by the server? 9
2.6 What users will be allowed to access the network services? 10
2.6.1 Web server 10
2.6.2 e-mail 11
2.6.3 ssh 11
2.7 What users will be allowed to log in and what privileges will they have? 12
2.8 What are you securing this system against? 12
2.9 How important or business critical is the server and its data? 13
2.10 Is the server directly exposed to the Internet? 13
2.11 What risks can reasonably be left in place? 14
3 Steps to Install and Harden the Server 15
3.1 Fedora RedHat installation 15
3.1.1 Test the CD media 15
3.1.2 Setup Options 15
3.1.3 Package Group Selection 17
3.2 Installation of apt-get 18
3.2.1 Upgrading current system 19
3.2.2 Maintenance of current packages 20
3.3 Locking down the server 20
3.3.1 Removing remote id 20
3.3.2 Limit system resource usage 21
3.3.3 Making sure root logs out 21
3.3.4 Limiting the consoles on the server 21
3.3.5 Removing unnecessary RPMs 22

3.3.6 Removing users 22
3.3.7 Turning services off 24
3.3.8 TCP Wrappers 25
3.3.9 Securing crontab 26
4 Permissions 26
4.1.1 Removing SUID and GUID from programs 26
4.1.2 Root permission 28
4.1.3 Group permission 28
4.1.4 Setting up sudo 29
4.2 Setting up services 30
4.2.1 SSH 30
4.2.2 Named 30
4.2.3 Sendmail 33
4.2.4 Securing MySQL 35
4.2.5 Configuring apache 36
4.2.6 Installing nessus 40
4.2.7 Installing Webmin 41
5 Design/Implement Ongoing Maintenance Procedures 43
5.1 System update 44
5.2 System Backup 44
6 Check Configuration 45
6.1 NMAP - Port scan 45
6.2 Run Nessus to check for any security vulnerabilities 45
6.3 Test Login 48
6.4 Verify system processes 49
6.5 Check that the chattr attribute works 49
7 Appendix 50
7.1 Partitioning 50
7.2 Bastille configuration file 50
7.3 Nessus scan results 52
7.4 Sendmail.conf 58

2 Server Specification and Risk Mitigation Plan

2.1 What role does this server fulfill?

The server is going to be a private server providing a limited number of users access to e-mail as well as hosting websites. These instructions are for a personal Linux server which was intended for private use only. However the way the system is configured is no different from a server located at an ISP with a high volume of traffic.

The services of the server are going to be used on a regular by several people. These services include e-mail, a web server and a domain name server. For these users the accounts for e-mail and their own web space will be used on a daily basis. Therefore the system needs to be set up and maintained in a manor which limits the disruption to the users.

Since the server is shared by several people it is only logical that the task of administration should also be shared amongst them. This means that several people will need to have shell access to the machine to enable them to configure the various services. Most of the users only use it for configuring one service, such as setting up DNS, and therefore really do not need full access to the machine (root).

For this reason the server will be set up in a role based environment. With the use of sudo the users can have just the right amount of access without being able to break the system or get access to files which they should not be able to.

2.2 What are the hardware requirements?

The server is not located in a server room and still needs to run 24/7. In order to be able to meet these requirements the need for a fanless server was apparent. After doing intense research on the Internet the Hush MiniITX, http://Hushtechnologies.net, server was the perfect option.

The quality of the server becomes apparent by the first time you open the box and the performance is meets and exceeds all the requirements of the system.

- **CPU**: 1 Ghz
- **RAM:** 1 GB
- **HD:** 120 MB
- **Optical drive:** CD-RW/DVD

2.3 What is the specific version of the operating system?

I choose RedHat Fedora core 1.0 for the installation.
http://fedora.redhat.com/download/

2.4 What third party software will be used?

There will be several tools used that do not come bundled up with the Fedora OS.

Name	Description	Version
Apt-get	Apt-get is a very handy tool for installing rpm packages. A port of Debian's apt tools for	apt-0.5.15cnc1-0.fdr.3.1.i386.rpm http://apt.freshrpms.net/

	RPM based distributions. It provides the apt-get utility that provides a simpler, safer way to install and upgrade packages. APT features complete installation ordering, multiple source capability and several other unique features.	
Bastille	Bastille is a tool for hardening the server. Bastille is a system hardening / lockdown program which enhances the security of a Unix host. It configures daemons, system settings and firewalls to be more secure. It can shut off unneeded services and r-tools, like rcp and rlogin, and helps create "chroot jails" that help limit the vulnerability of common Internet services like Web services and DNS.	Bastille-2.1.1-1.0.i386.rpm http://www.bastille-linux.org/
Nessus	Nessus is a penetration testing tool. With it you can check if your server has known security risks.	2.0.10a http://www.nessus.org/download.html
Webmin	A web-based administration interface for Unix systems. Using Webmin you can configure DNS, Samba, NFS, local/remote filesystems and more using your web browser.	webmin-1.130-1 http://www.webmin.com/download.html

2.5 What services will be offered by the server?

The following services will need to be accessible by the users:

Service name	Description	Daemon
HTTP	Basic web server	Apache
HTTPS	Secure web server. Used for accessing web mail and websites that require login	Apache
IMAP	IMAP server for accessing e-mail via remote tools such as MS Outlook	imap-2002d-3
IMAPS	Same as IMAP except that it uses SSL encryption for communication. I have to leave IMAP on because users found it very annoying that MS Outlook prompts that the certificate is not trusted. (Once I figure out how to prevent this without having to purchase a signed certificate ill turn IMAP off.)	imap-2002d-3
POP	Same as above	Same as above
POPs	Same as above	Same as above
SSH	This package contains the secure shell daemon (sshd). The sshd daemon allows SSH clients to securely connect to your SSH server.	openssh-server-3.6.1p2-19
MySQL	MySQL is a multi-user, multi-threaded SQL database server. MySQL is a client/server implementation consisting of a server daemon (mysqld) and many different client programs and libraries. This package contains the MySQL client	mysql-server-3.23.58-4

	programs, the client shared library, and generic MySQL files.	
PostgreSQL	The postgresql-server package includes the programs needed to create and run a PostgreSQL server, which will in turn allow you to create and maintain PostgreSQL databases. PostgreSQL is an advanced Object-Relational database management system (DBMS) that supports almost all SQL constructs, including transactions, subselects, and user-defined types and functions. You should install postgresql-server if you want to create and maintain your own PostgreSQL databases and/or your own PostgreSQL server. You also need to install the postgresql package.	postgresql-server-7.3.4-11

2.6 What users will be allowed to access the network services?

2.6.1 Web server

The web server will be accessible to anyone on the internet. It will host mainly simple web pages that consist out of static HTML. However there will be a couple of sites that will require a more complex set up; both of which will run on PHP. One virtual host will run the web e-mail client, allowing users to access their e-mail from around the world. This server is going to be SSL enabled. Since the other services do not send over personal information nor do they ask the person for username and a password placing them on a SSL host is not necessary.

2.6.2 e-mail

There will be several people who will use the server as their primary e-mail account. This places several conditions on the server regarding up time and usability. Most users will be using MS Outlook to access their IMAP account while others prefer to use POP out of archival issues (they seldom have a stable internet connection and really need to copy all the files to their local machine to work on them). In both of these cases I had to compromise between security and user friendliness. I have installed both the IMAPs and the POPs servers, however, most users find the pop-up-screen very annoying so they do not want to use it and there are not enough users nor secure information to really validate the need to purchase a certificate from a provider.

All outgoing e-mail to the server has to be authenticated. This makes the SMTP server more secure and prevents SPAMers from using the machine as a jumping point. Here again limitations in MS Outlook prevented us from using a secure method of transmitting the password. They are done via the LOGIN option in the SMTP.

To further reduce the risk of having user credentials reviled which could compromise the system. E-mail accounts will be separate from the accounts that need to have shell access.

2.6.3 ssh

Only a selected few users should be able to access the shell of the server. It would be nice to limit the access to a selected range of ip-addresses but this would conflict with the usability of the system. Therefore I only limited access to the shell to a select few users, all of which actually help administer services. None of the accounts that are allowed to log into the machine will be used for accessing e-mail. This eliminated the risk for the password being sniffed when used with unsafe protocols such as IMAP or POP.

2.7 What users will be allowed to log in and what privileges will they have?

Users will only be allowed to access the server if they have a specific role on the server, for instance if a user is in the webmaster group or the dnsmaster group. These users will be limited to modifying the configuration files that they are allowed to access. They will need to use sudo to gain the privileges they need to restart services.

Root will only be allowed to log in if the correct ssh-key is used for logging into the machine. This is for doing automated backups of the file system to another host.

None of the e-mail accounts will have access to the server. A second account will be created for those users that require both e-mail and shell access. This protects these accounts from being sniffed do to the use of unsafe protocols such as IMAP or POP.

2.8 What are you securing this system against?

There are two methods that the server can be compromised in. The first is an attack from the internet where services would need to be compromised and the second is from users that can access the server directly.

In the first case the types of attacks are oriented towards the services that are accessible to the attacker. This includes:

- Denial of service
- Buffer overflows
- Password hacking

To litigate the chance of these attacks most of the services should be configured correctly with the latest version of the software. I don't view the denial of service attack as a high risk situation because the server is connected to a cable modem and has more than enough resources available. Therefore the attacker would really need to exploit a flaw in the code to (i.e. buffer overflow) to do any damage on the server. The third attack,

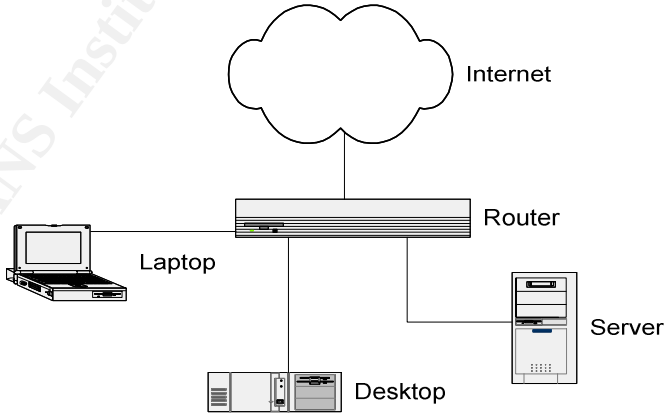
directed at users passwords has the highest risk. Here only training the user can assist in the prevention that this might happen. However, since the users require access to their e-mail via an unsecured channel it is very easy to a hacker to sniff their passwords from the network. However, luckily all users that have permission to access the shell do not have login permissions and the extent of the damage is limited to their personal e-mail accounts.

The other threat is from users on the system. These users will have limited permissions on the system and cannot alter configuration files on the server. If possible I tried to confine daemons in a chrooted environment to further limit the extent of the security risk to the server.

2.9 How important or business critical is the server and its data?

The server primarily intended for private use. Therefore even if the server is down for one day the damage will not be substantial. Most of the data on the server is also not critical and can be recreated at any time. These two factors allow a little more slack in maintaining the data integrity of the server and provide in a more flexibility in choosing the backup solution.

2.10 Is the server directly exposed to the Internet?



The network is connected via a HUB/Router. All the machines on the internal network have a private ip-address and only the services required from the outside network are

forwarded directly to the server. The router is masquerading the network to towards the internet and also has a firewall installed on it.

The firewall on the router blocks all traffic that is trying to get into the internal network and since it is running NAT only requests that were initiated from within the private network will be forwarded back into the internal network.

This setup allows for the use of services that are only intended for internal use such as tightVNC while still remaining secure from attacks from the internet. In addition the use of other devices/services such as printers for file sharing can be performed between the laptop and the desktop without being at risk from out side attacks.

2.11 What risks can reasonably be left in place?

The only risk that has to be accepted is that the passwords for the e-mail accounts can be compromised. This risk is real in every situation and there is also no fool proof method to detect or prevent this from happening. The risk is high because most of these users do not use the HTTPS connection to read their mail, rather they use the unsecured IMAP / POP /SMTP services to access their mail.

Here the users have to remain alert and keep an eye on their account to make sure that no harmful action has taken place. In the case of a compromise their password will be reset.

3 Steps to Install and Harden the Server

3.1 Fedora RedHat installation

I choose RedHat Fedora core 1.0 for the installation.
http://download.fedora.redhat.com/pub/fedora/linux/core/1/i386/iso/

The server was not installed while connected on the network. This was not as much as a security risk as the need to connect it to the keyboard and monitor. Since the server is in the private network the risk of having a vulnerable server online are minimized. However, if the server were connected directly to the internet I would strongly advise not having it connected to the internet until the system is fully secured.

3.1.1 Test the CD media

Before Installation you should check the media to make sure that they are ok.
When prompted select the option to test the CD media.

If the media is ok you should see the following results:
The media check of the image:
Fedora Core 1 disc 1
Is complete, and the result is: PASS.
It is OK to install from this media.

3.1.2 Setup Options

Question/Option	Explanation	Action
Welcome screen	Welcome screen	Select: Next
Language	This is the Language that will be used	Select: English

	during the installation	
Keyboard	This is the keyboard layout	Select: U.S. English
Mouse	This is the mouse driver	Select: Wheel mouse (PS/2)
Installation Type	This screen gives the user several presets for the configuring the system. They range from Everything to Custom.	Select: Custom
Partitioning	You can select manual or automatic partitioning.	See Appendix A for details.
Boot loader	This screen allows you to configure the boot loader options. In most cases the default should be good enough. Selecting a password for Grub will make it harder for someone to reset the root password if they have physical access to the machine.	Select: Grub boot loader to be installed on /dev/hda Select: "Use a boot loader password" and enter a password in the prompt.
Network Configuration	Here you can setup the network. I would recommend doing this because once the system reboots you can configure it via ssh.	Input your network settings
Firewall Configuration	I have disabled the firewall which based on iptables. I don't like the way it configures the system and is not flexible enough for my personal needs I will replace it with other scripts in the future	Select: disabled
Additional Language Support	This option will install additional man pages on your system. However, in my past experience it is best to stick to	Select: English (USA)

	one. It saves space and since all of the users know English there is no need for them.	
Time Zone	Selecting the time zone is important for keeping your systems logs in correct time. You should also select the NTP for synchronization to ensure that the server has the right time. If you need to change your timezone later on you can do this in the file /etc/sysconfig/clock	Select: America/New York – Eastern time
Enter root password	The root password should be hard to crack. This password should have at least 8 characters and mix in special characters. Also, you should not use the same password for a user account.	Enter password

3.1.3 Package Group Selection

I selected the following packages to be installed:

Package	Selected	Description
X Windows System	X	
Gnome Desktop Environment		
KDE Desktop Environment	X	This is personal preference. I just like the interface a little more than Gnome
Editors	X	Just wanted to have vim enhanced an emacs (since I will be programming from the shell)
Engineering * Scientific		
Graphical Internet	X	Mozilla webbrowser etc.

Text-based Internet	X	I like tools like links and ncftp and lynx
Office Productivity		
Sound and Video		
Authoring and Publishing		
Graphics		
Games and Entertainment		
Server Configuration Tools		
Web Server	X	With all the features selected
Mail Server	X	
Windows File Server	X	Samba is only used for ad-hoc situations and will not be started on startup.
DNS Name Server	X	
FTP Server		
SQL Database Server	X	Also select MySQL
News Server		
Network Servers	X	Here I selected VNC, to make administration easier
Development Tools		
Kernel Development		
X Software development		
GNOME Software Development		
KDE Software Development		
Administration Tools		
System Tools	X	Ethereal, NMAP
Printing Support		

3.2 Installation of apt-get

There are many tools that allow for package installation and updates. My personal preference is apt-get. The tool has the nice feature of handling all the dependencies

required for an installation. In addition it checks the signature of each rpm before it tries to install them.

```
wget http://download.fedora.us/fedora/fedora/1/i386/RPMS.stable/apt-
0.5.15cnc1-0.fdr.3.1.i386.rpm
```

Install apt get with the following command.

```
rpm -Uvh apt-0.5.15cnc1-0.fdr.3.1.i386.rpm
```

3.2.1 Upgrading current system

You need to update the local system with the upgrade option. This will also download the GPG keys for ensuring that the rpms are in good condition.

```
[root@hush root]# apt-get update
You don't seem to have one or more of the needed GPG keys in your RPM
database.
Importing them now...
Get:1 http://download.fedora.us fedora/1/i386 release [2494B]
Get:2 http://macromedia.mplug.org fedora/1 release [505B]
Fetched 2999B in 4s (600B/s)
Get:1 http://download.fedora.us fedora/1/i386/os pkglist [1445kB]
Get:2 http://macromedia.mplug.org fedora/1/macromedia pkglist [1036B]
Get:3 http://macromedia.mplug.org fedora/1/macromedia release [129B]
Get:4 http://macromedia.mplug.org fedora/1/macromedia srclist [768B]
Get:5 http://download.fedora.us fedora/1/i386/os release [124B]
Get:6 http://download.fedora.us fedora/1/i386/updates pkglist [285kB]
Get:7 http://download.fedora.us fedora/1/i386/updates release [129B]
Get:8 http://download.fedora.us fedora/1/i386/stable pkglist [280kB]
Get:9 http://download.fedora.us fedora/1/i386/stable release [122B]
Get:10 http://download.fedora.us fedora/1/i386/os srclist [157kB]
Get:11 http://download.fedora.us fedora/1/i386/updates srclist [18.1kB]
Get:12 http://download.fedora.us fedora/1/i386/stable srclist [49.3kB]
Fetched 2237kB in 16s (136kB/s)
Reading Package Lists... Done
Building Dependency Tree... Done
```

Then use the update option to download and install the out-of date rpms.

```
root@hush root]# apt-get upgrade
Reading Package Lists... Done
Building Dependency Tree... Done
The following packages will be upgraded
  XFree86 XFree86-100dpi-fonts XFree86-75dpi-fonts XFree86-Mesa-libGL
  XFree86-Mesa-libGLU XFree86-base-fonts XFree86-font-utils XFree86-libs
  XFree86-libs-data XFree86-tools XFree86-truetype-fonts XFree86-twm
  XFree86-xauth XFree86-xdm XFree86-xfs bash ethereal ethereal-gnome
foomatic
  gaim gdm ghostscript gimp-print glibc glibc-common gnome-libs gnupg
gphoto2
  grep hpijs httpd httpd-manual hwdata initscripts iptables kernel-
pcmcia-cs
```

```
  lftp libxml2 libxml2-python mailman mod_python mod_ssl mozilla mozilla-
mail
  mozilla-nspr mozilla-nss mutt net-snmp net-snmp-utils nscd nss_ldap
pam_krb5
  pango perl-DateManip php php-imap php-ldap php-mysql php-odbc php-pgsql
  postgresql postgresql-libs postgresql-server procps redhat-config-
packages
  redhat-config-printer redhat-config-printer-gui rhn-applet rsync samba
  samba-client samba-common sed slocate spamassassin vnc vnc-server xchat
78 upgraded, 0 newly installed, 0 removed and 0 not upgraded.
Need to get 164MB of archives.
After unpacking 22.4MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

3.2.2 Maintenance of current packages

In order to make sure that I always have the latest packages installed I created a cron script that updates the RPMs on a monthly basis. I called a script /etc/cron.monthly/apt-get.sh and made it executable.

```
#!/bin/sh

APT=/usr/bin/apt-get

echo "UPDATING PACKAGES"
${APT} update
${APT} -y upgrade
```

3.3 Locking down the server

3.3.1 Removing remote id

The remote id is displayed before login in to the server via the console. I am apposed to displaying the version of the system here. Anther option is to put a security warning on the login.

```
[root@hush etc]# cat /etc/issue
Fedora Core release 1 (Yarrow)
Kernel \r on an \m
```

```
[root@hush etc]# rm -f /etc/issue /etc/issue.net
```

3.3.2 Limit system resource usage

Limiting the system to certain resources can help in fighting denial of service attacks. In addition it will make the systems load be more equalized for all users.

Core dumps are not used often anymore for debugging the system therefore there is no need to have them clutter up “random” places of the HD. Core dumps can also prose a security risk to those that have access to them. With the strings command the attacker could potentially find out information that would otherwise not be accessible to them such as the contents of the password file. Core dumps contain all the information that the originating process had read.

```
# turn off core dumps
*               hard    core            0
# limit the user processes to 120
*               soft    nproc           120
*               hard    nproc           120
```

3.3.3 Making sure root logs out

A high risk is that during maintenance root will be accidentally left logged in on the shell. To prevent this add the following line to /etc/profile

```
export TMOUT=3600
```

3.3.4 Limiting the consoles on the server

When an item is not needed it is a good idea to turn it off. This holds true for the terminals on the server. I have turned off all consoles except one on the server.

```
cat /etc/securetty
console
vc/1
#vc/2
#vc/3
#vc/4
#vc/5
#vc/6
#vc/7
#vc/8
#vc/9
#vc/10
#vc/11
tty1
```

```
#tty2
#tty3
#tty4
#tty5
#tty6
#tty7
#tty8
#tty9
#tty10
#tty11
```

3.3.5 Removing unnecessary RPMs

Once I am sure that I will never need an rpm I will remove it entirely from the system. This would give would-be hackers even less to work on if they manage to get a shell on the system.

```
rpm -e webalizer
rpm -e rsh
rpm -e squid
rpm -e nfs-utils
rpm -e at
rpm -e nscd nss_ldap
```

3.3.6 Removing users

After the system was installed there are a lot of users that are unnecessary for the operation of the system. It is best to remove these users to reduce the risk of unauthorized access.

User	Comments
root:x:0:0:root:/root:/bin/bash	Admin user
bin:x:1:1:bin:/bin:/sbin/nologin	Needed for some scripts/permissions
daemon:x:2:2:daemon:/sbin:/sbin/nologin	Needed for some init scripts
adm:x:3:4:adm:/var/adm:/sbin/nologin	Delete
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin	Printer
sync:x:5:0:sync:/sbin:/bin/sync	
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown	Needed for initscripts
halt:x:7:0:halt:/sbin:/sbin/halt	Needed for

	initscripts
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin	Sendmail
news:x:9:13:news:/etc/news:	Delete
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin	Delete
operator:x:11:0:operator:/root:/sbin/nologin	Delete
games:x:12:100:games:/usr/games:/sbin/nologin	Delete
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin	Delete
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin	Delete
nobody:x:99:99:Nobody:/:/sbin/nologin	Delete
rpm:x:37:37::/var/lib/rpm:/sbin/nologin	
vcsa:x:69:69:virtual console memory owner:/dev:/sbin/nologin	
nscd:x:28:28:NSCD Daemon:/:/sbin/nologin	
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin	
rpc:x:32:32:Portmapper RPC user:/:/sbin/nologin	
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin	
nfsnobody:x:65534:65534:Anonymous NFS User:/var/lib/nfs:/sbin/nologin	
mailnull:x:47:47::/var/spool/mqueue:/sbin/nologin	Procmail
smmsp:x:51:51::/var/spool/mqueue:/sbin/nologin	Procmail
pcap:x:77:77::/var/arpwatch:/sbin/nologin	
apache:x:48:48:Apache:/var/www:/sbin/nologin	Apache
squid:x:23:23::/var/spool/squid:/sbin/nologin	
webalizer:x:67:67:Webalizer:/var/www/html/usage:/sbin/nologin	Delete
dbus:x:81:81:System message bus:/:/sbin/nologin	
xfs:x:43:43:X Font Server:/etc/X11/fs:/sbin/nologin	
named:x:25:25:Named:/var/named:/sbin/nologin	
ntp:x:38:38::/etc/ntp:/sbin/nologin	
desktop:x:80:80:desktop:/var/lib/menu/kde:/sbin/nologin	
gdm:x:42:42::/var/gdm:/sbin/nologin	
dovecot:x:97:97:dovecot:/usr/libexec/dovecot:/sbin/nologin	Imap
postfix:x:89:89::/var/spool/postfix:/sbin/nologin	Postfix
mailman:x:41:41:GNU Mailing List Manager:/var/mailman:/sbin/nologin	Mailman
mysql:x:27:27:MySQL Server:/var/lib/mysql:/bin/bash	Mysql
postgres:x:26:26:PostgreSQL Server:/var/lib/pgsql:/bin/bash	Postgresql

3.3.7 Turning services off

In most cases the best option for securing unused services is to actually uninstall the various services. However, on the current setup I do not perceive the security risk as high enough to remove them completely from the system. I might opt to use them later on the internal network.

There are too many services running on the server. If a service is not used it is always best to turn it off. Not only does it increase the security, it also frees up additional resources for the server.

```
[root@hush mail]# chkconfig --list|grep on
syslog          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
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
saslauthd       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
acpid           0:off   1:off   2:off   3:on    4:on    5:on    6:off
irqbalance      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
microcode_ctl   0:off   1:off   2:on    3:on    4:on    5:on    6:off
smartd          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
spamassassin    0:off   1:off   2:on    3:on    4:on    5:on    6:off
sendmail        0:off   1:off   2:on    3:on    4:on    5:on    6:off
crond           0:off   1:off   2:on    3:on    4:on    5:on    6:off
httpd           0:off   1:off   2:on    3:on    4:on    5:on    6:off
messagebus      0:off   1:off   2:off   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
named           0:off   1:off   2:on    3:on    4:on    5:on    6:off
ntpd            0:off   1:off   2:off   3:on    4:off   5:on    6:off
mysqld          0:off   1:off   2:on    3:on    4:on    5:on    6:off
webmin          0:off   1:off   2:on    3:on    4:off   5:on    6:off
        imap:   on
        imaps:  on
        ipop3:  on
        pop3s:  on
```

These are the services that I turned off:

```
chkconfig portmap off
chkconfig isdn off
chkconfig cups off
chkconfig rhnsd off
chkconfig netfs off
chkconfig kudzu off
chkconfig autofs off
chkconfig gpm off
chkconfig nfslock off
chkconfig sgi_fam off
```

3.3.8 TCP Wrappers

TCP wrappers are an easy to configure access restrictions to the various services. Rather than accidentally misconfigure a service and have it exposed a locked down approach to TCP wrappers is not only quick to set up, but it forces you to specifically allow the services you want to have accessed from a remote machine.

First lets take away all the rights with the ALL:ALL rule in the hosts.deny file

```
[root@hush root]# cat /etc/hosts.deny
#
# hosts.deny    This file describes the names of the hosts which are
#               *not* allowed to use the local INET services, as decided
#               by the '/usr/sbin/tcpd' server.
#
# The portmap line is redundant, but it is left to remind you that
# the new secure portmap uses hosts.deny and hosts.allow.  In particular
# you should know that NFS uses portmap!
ALL:ALL
```

Now we need to allow access to the services we want to be able to access from the outside.

```
[root@hush root]# cat /etc/hosts.allow
#
# hosts.allow   This file describes the names of the hosts which are
#               allowed to use the local INET services, as decided
#               by the '/usr/sbin/tcpd' server.
#
sshd:ALL
imapd:ALL
ipop3d:ALL
sendmail:ALL
```

TCP Wrappers is an easy way to control access to services. For instance if I were to decide to limit SSH access to the private subnet you just need to switch the line sshd:ALL to sshd:192.168.1.0/24

3.3.9 Securing crontab

By default any user can add a crontab entry. To limit this privilege to a few selected users you need to create the cron.allow file.

```
[root@hush etc]# touch /etc/cron.allow
```

If you want to allow a user to add their own crontab entry you can write their name in the file. Once they have installed the crontab and they are not expected to make changes often you can take out the entry in crontab. This will allow the crontab to continue to run and is more secure since the user cannot edit the crontab anymore.

4 Permissions

4.1.1 Removing SUID and GUID from programs

```
find / -type f \( -perm -04000 -o -perm -02000 \) -exec ls -l {} \;
```

remove the suid with the following command:

```
chmod a-s filename
```

Program	Comment
/var/mailman/cgi-bin/edithtml	Leave
/var/mailman/cgi-bin/admin	Leave
/var/mailman/cgi-bin/admindb	Leave
/var/mailman/cgi-bin/confirm	Leave
/var/mailman/cgi-bin/create	Leave
/var/mailman/cgi-bin/listinfo	Leave
/var/mailman/cgi-bin/options	Leave
/var/mailman/cgi-bin/private	Leave
/var/mailman/cgi-bin/rmlist	Leave

/var/mailman/cgi-bin/roster	Leave
/var/mailman/cgi-bin/subscribe	Leave
/var/mailman/mail/mailman	Leave
/usr/X11R6/bin/XFree86	Leave
/usr/sbin/usernetctl	Remove
/usr/sbin/userhelper	Remove
/usr/sbin/lockdev	Leave
/usr/sbin/utempter	Leave
/usr/sbin/userisdnctl	Remove
/usr/sbin/sendmail.sendmail	Leave
/usr/sbin/suexec	Leave
/usr/sbin/gnome-pty-helper	Leave
/usr/sbin/postdrop	Leave
/usr/sbin/postqueue	Leave
/usr/bin/chage	Remove
/usr/bin/gpasswd	Remove
/usr/bin/wall	Remove
/usr/bin/chfn	Remove
/usr/bin/chsh	Remove
/usr/bin/newgrp	Remove
/usr/bin/write	Remove
/usr/bin/passwd	Leave
/usr/bin/lockfile	Leave
/usr/bin/slocate	Leave
/usr/bin/at	Leave
/usr/bin/sudo	Leave
/usr/bin/crontab	Leave
/usr/bin/lppasswd	Remove
/usr/bin/desktop-create-kmenu	Leave
/usr/lib/vte/gnome-pty-helper	Leave
/usr/libexec/openssh/ssh-keysign	Leave

/bin/ping	Remove
/bin/ping6	Remove
/bin/traceroute6	Remove
/bin/mount	Remove
/bin/umount	Remove
/bin/su	Leave
/bin/traceroute	Remove
/sbin/pam_timestamp_check	Leave
/sbin/pwdb_chkpwd	Leave
/sbin/unix_chkpwd	Leave
/sbin/netreport	Leave

4.1.2 Root permission

Services on the server should be limited to users who have permission to do so. I have set up sudo rights for the various groups so they will not need to access any of the init script directly. Hence the following is a secure method of even preventing others to see the services they should not use:

```
chmod -R 700 /etc/init.d
```

Since all the crond.* directories are run as the user root there is no need to make them readable to other users.

```
chmod -R 700 /etc/cron.*
```

4.1.3 Group permission

For now I only have two types of administrators, one for the web server the other for the dns server. I want to be able to allow the various groups to administer their server and load the new configuration within them.

```
chgrp –R webmaster /etc/httpd
chmod 770 /etc/httpd
```

```
chgrp dnsmaster /var/named
chown named /var/named # this will allow the named to enter into the
directory
chmod 550 /var/named
```

4.1.4 Setting up sudo

To really make the role based administration work the admins need to be able to restart their services and look into the logs. The following configuration of the sudo file will enable them to do that.

As an added measure I did not add the NOPASSWD option to their entries. This means that when they run the sudo command they will be prompted for the password. They do not need to re-enter the password for the next 5 min which makes the system easier to use.

```
cat /etc/sudoers |grep -v "#"

Cmnd_Alias TAIL=/usr/bin/tail
Cmnd_Alias NAMED=/sbin/service named *
Cmnd_Alias HTTPD=/sbin/service httpd *

root    ALL=(ALL) ALL
%webmaster     ALL=(ALL)       HTTPD,TAIL
%dnsmaster      ALL=(ALL)       NAMED,TAIL
```

4.2 Setting up services

4.2.1 SSH

The ssh server is essential for any UNIX/Linux machine. In order to prevent un-authorized users from logging onto the machine the default values of the ssh daemon need to be altered.

By default the ssh daemon allows root to log into the server as well as any other valid user. This poses a problem since there are several users who will only be using the server for their e-mail. They should don't need to have access to the machine. The solution is quite simple.

This configuration limits the protocol to version 2 since the version 1 has some security issues associated with it. The root login was limited to ssh-keys, this makes it possible to create backups using rsync via shh. I turned of X11 forwarding because it will not be needed in my environment and therefore it would only pose a risk. To limit the users allowed onto the machine I created specific roles for the various users. At this stage only two are defined webmaster and the dnsmaster. These groups also have sudo rights for the various services. Therefore any new user just added to the system will not be able to access the server unless the administrator adds them to either one of these groups. The only items that were left unchanged were the logging facility and the sftp service.

```
# cat /etc/ssh/sshd_config | grep -v "#"
Protocol 2
SyslogFacility AUTHPRIV
X11Forwarding no
AllowGroups webmaster dnsmaster
Subsystem       sftp    /usr/libexec/openssh/sftp-server
```

4.2.2 Named

The Bind daemon has been known in the past for its various security flaws. That is why the newer version of the Bind daemon already comes with the option to run in a chrooted

environment. Named was already chrooted after the installation. This is a little confusing because the configuration files could still be found outside of the chrooted environment.

```
[root@hush root]# cat /etc/sysconfig/named
# Currently, you can use the following options:
# ROOTDIR="/some/where"  --  will run named in a chroot environment.
#                        you must set up the chroot environment before
#                            doing this.
# OPTIONS="whatever" -- These additional options will be passed to named
#                   at startup. Don't add -t here, use ROOTDIR instead.
ROOTDIR=/var/named/chroot
```

Next we need to make sure that the service is actually started by default.

```
[root@hush root]# chkconfig named on
```

Next I am changing the group permissions of the named folder. This will allow the proper use of the group dnsmaster where all users will be able to modify the configuration files without compromising the rest of the system.

```
# find /var/named -type d -group dnsmaster -exec chmod g+s \{} \;
```

Hereafter you should remove the configuration files out side of the chrooted environment. They will only lead to confusion when configuring the server.

```
rm -rf /var/named/localhost.zone
rm -rf /var/named/named.ca
rm -rf /var/named/named.local
rm -rf /var/named/slaves/
```

Create a symbolic link from the named.conf file outside of the chroot to the one inside of the chroot.

```
rm -f /etc/named.conf
ln -s /var/named/chroot/etc/named.conf /etc/named.conf
```

Since this Bind daemon is going to host several sites it is important to make sure the configuration is done correctly. I.e. the statement for allow-transfer should contain all the

ip-address of the slave servers. If you don't do this your named will allow zone transfers from any host resulting in providing the would-be hacker with an in-depth look at your network setup.

I like the logging statement because it gives me a better impression of what the system is doing all the time. Since the server is running in a chrooted environment it cannot log to syslog, hence it will write to a file in the chrooted environment.

```
options {
        directory "/var/named";
        allow-transfer { XXX.XXX.XXX.XXX; };
        /*
         * If there is a firewall between you and nameservers you want
         * to talk to, you might need to uncomment the query-source
         * directive below.  Previous versions of BIND always asked
         * questions using port 53, but BIND 8.1 uses an unprivileged
         * port by default.
         */
         query-source address * port 53;
};

logging {
        channel query_logging {
                // syslog named.query;
                file "/var/log/named_querylog"
                versions 3 size 50M;
                print-time yes; // timestamp log entries
        };

        category queries {
                query_logging;
        };
};
//
// a caching only nameserver config
//
controls {
        inet 127.0.0.1 allow { localhost; } keys { rndckey; };
};
zone "." IN {
        type hint;
        file "named.ca";
};

zone "localhost" IN {
        type master;
        file "localhost.zone";
        allow-update { none; };
};

zone "0.0.127.in-addr.arpa" IN {
        type master;
        file "named.local";
        allow-update { none; };
};
```

```
/**
 * These domains are hosted on the server
 */

zone "servername.com"{
        type master;
        file "named. servername.com";
        allow-transfer { XX.XXX.XXX.XXX; };
        notify yes;
};
```

Here is a dns configuration

```
cat /var/named/chroot/var/named/named.servername.com
$TTL 86400
@       3600    IN      SOA     servername.com.
postmaster.servername.com. (
                        2004032701 ; serial
                        36000 ; refresh
                        3600 ; retry
                        604800 ; expire
                        3600 ; default_ttl
                        )
@               IN      TXT     "servername.com "
@               IN      NS      ns1.servername.com.
@               IN      NS      ns2.servername.com.
@               IN      MX      0       mail.servername.com.
ns1             IN      A       xxx.xxx.xxx.xxx
ns2             IN      A       xxx.xxx.xxx.xxx
@               IN      A       xxx.xxx.xxx.xxx
host           IN      A       xxx.xxx.xxx.xxx
www             IN      CNAME   host
*               IN      CNAME   host
```

4.2.3 Sendmail

Setting up Sendmail can be very confusing and in many cases miss-configurations result in security risks.

To prevent people from Relaying over the server you need to install the cyrus packages. However, MS Outlook only supports the plain authentication. I know this sends the passwords in an easy to crack method, but here is a point where usability has to proceed functionality. Most of the Cyrus packages are installed by default, except the one that is required for Office.

```
apt-get install cyrus-sasl-plain
```

Next you need to link the SMTP to the pluggable authentication module (PAM) by creating the according file in the /etc/pam.d directory.

```
cat /etc/pam.d/smtp
#%PAM-1.0
auth       required     pam_stack.so service=system-auth
account    required     pam_stack.so service=system-auth
```

Thereafter you need to add the various relays into the file /etc/mail/access. These hosts will be able to send e-mail without having to authorize. In my case I am forwarding my e-mails to another host. This remote host needs to have my ip in its access table so I can relay over it.

```
cat /etc/mail/access
# Check the /usr/share/doc/sendmail/README.cf file for a description
# of the format of this file. (search for access_db in that file)
# The /usr/share/doc/sendmail/README.cf is part of the sendmail-doc
# package.
#
# by default we allow relaying from localhost...
localhost.localdomain            RELAY
localhost                        RELAY
127.0.0.1                        RELAY
```

4.2.3.1 Changes to Sendmail.mc

I am using a cable modem for my internet set up. Because it is easy to spam from a connection like mine I found out that I could not send e-mails to certain addresses – the range was on the block list. Therefore I had to set up my mail server to relay over another server. The remote server needs to have my ip-address in its /etc/mail/access file to allow relaying.

```
dnl # Uncomment and edit the following line if your outgoing mail needs
to
dnl # be sent out through an external mail server:
dnl #
define(`SMART_HOST',`xxx.xxx.xxx.xxx')
```

The following is to allow authentication for SMTP requests. Only users with a valid user/password are allowed to send e-mail from the server this protects me from someone using my server as a spam-relay.

```
dnl #
dnl # The following allows relaying if the user authenticates, and
disallows
dnl # plaintext authentication (PLAIN/LOGIN) on non-TLS links
dnl #
dnl define(`confAUTH_OPTIONS', `A p')dnl
dnl #
dnl # PLAIN is the preferred plaintext authentication method and used by
dnl # Mozilla Mail and Evolution, though Outlook Express and other MUAs
do
dnl # use LOGIN. Other mechanisms should be used if the connection is not
dnl # guaranteed secure.
dnl #
TRUST_AUTH_MECH(`EXTERNAL DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dnl
define(`confAUTH_MECHANISMS', `EXTERNAL GSSAPI DIGEST-MD5 CRAM-MD5 LOGIN
PLAIN')
```

Only allowing relaying based on the MX record prevents some spam from getting on the server.

```
dnl # We strongly recommend not accepting unresolvable domains if you
want to
dnl # protect yourself from spam. However, the laptop and users on
computers
dnl # that do not have 24x7 DNS do need this.
dnl FEATURE(`accept_unresolvable_domains')dnl
dnl #
FEATURE(`relay_based_on_MX')dnl
dnl #
dnl # Also accept email sent to "localhost.localdomain" as local email.
dnl #
LOCAL_DOMAIN(`localhost.localdomain')dnl
```

4.2.4 Securing MySQL

The MySQL server will be used on several websites. There is not any need in the near future where direct connection to MySQL from a foreign host will be necessary. Therefore binding the server to local host is a good idea.

```
cat /etc/my.cnf
[mysqld]
bind-address=127.0.0.1
datadir=/var/lib/mysql
```

```
socket=/var/lib/mysql/mysql.sock

[mysql.server]
user=mysql
basedir=/var/lib

[safe_mysqld]
err-log=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
```

4.2.5 Configuring apache

Apache is next to sendmail the most used service on the server. For this reason it has to be configured correctly to prevent any exploits to be run on the server. In order to prevent miss configurations I separated the virtualhosts from the rest of the configuration. So while the basic server setup is located in **/etc/httpd/conf/httpd.conf** most of the changes will happen in /**etc/httpd/conf.d/virtualhost.conf** and **/etc/httpd/conf.d/ virtualhost_ssl.conf** for ssl hosts.

The next step is to limit the information that the attacker can find out about the server. The following two configurations limit the information that apache sends to the remote host.

```
# Don't give away too much information about all the subcomponents
# we are running.  Comment out this line if you don't mind remote sites
# finding out what major optional modules you are running
ServerTokens Prod
```

```
# Optionally add a line containing the server version and virtual host
# name to server-generated pages (internal error documents, FTP directory
# listings, mod_status and mod_info output etc., but not CGI generated
# documents or custom error documents).
# Set to "EMail" to also include a mailto: link to the ServerAdmin.
# Set to one of:  On | Off | EMail
#
ServerSignature Off
```

Because I am using webdav to upload files to the server I changed the group of apache to web so that I can have both scp and webdav access to the server. I was not fond of the idea to place other users into the apache group therefore I changed the group of apache. Under apache 1.3.x I could set the user and group in the virtual host file, which would have been the more elegant solution, but sadly that does not work in apache 2.x anymore.

```
User apache
Group web
```

The following was a recommendation from Nessus. Without it the server is potentially vulnerable to redirect attacks which can deceive your users into submitting information on a foreign, untrusted host.

```
RewriteEngine on
RewriteCond %{REQUEST_METHOD} ^(TRACE|TRACK)
RewriteRule .* - [F]
```

Finally I need to have index.php as an index for several websites.

```
DirectoryIndex index.html index.php
```

4.2.5.1 Generating SSL keys

You can only use one SSL key per ip-address. I only have one external ip-address therefore I generated a key for the application that I use the most, namely webmail. The fist SSL key that apache finds in the configuration will be used for all virtualhosts, even when you specify another file.

I created a script that automatically runs all the commands you need in order to generate a proper SSL certificate and sign it too so you do not have to enter the password every time you need to restart the apache server. Place the file in the **/etc/httpd/conf** directory. The script takes the ServerName as its only parameter i.e. to run it use the following command: **/etc/httpd/conf/mkcerts.sh www.mysite.com** .

```
cat mkcerts.sh
#/bin/sh

SERVER_NAME=$1

echo "Generating Certificate for ${SERVER_NAME}"

echo "openssl genrsa -des3 -rand file1:file2:file3:file4:file5 -out
${SERVER_NAME}.key 1024"
```

```
openssl genrsa -des3 -out ${SERVER_NAME}.key 1024
```

```
echo "openssl rsa -in ${SERVER_NAME}.key -out ${SERVER_NAME}.pem"
openssl rsa -in ${SERVER_NAME}.key -out ${SERVER_NAME}.pem

echo "openssl req -new -key ${SERVER_NAME}.key -out ${SERVER_NAME}.csr"
openssl req -new -key ${SERVER_NAME}.key -out ${SERVER_NAME}.csr

echo "openssl x509 -req -days 360 -in ${SERVER_NAME}.csr -signkey
${SERVER_NAME}.key -out ${SERVER_NAME}.crt"
openssl x509 -req -days 360 -in ${SERVER_NAME}.csr -signkey
${SERVER_NAME}.key -out ${SERVER_NAME}.crt

echo "moving certs to folder"
mv ${SERVER_NAME}.crt ssl.crt/
mv ${SERVER_NAME}.csr ssl.csr/
mv ${SERVER_NAME}.key ssl.key/
mv ${SERVER_NAME}.pem ssl.prm/
```

Once the key has been generated insert the following lines to activate SSL on the server.

```
    SSLEngine on
    SSLCertificateFile /etc/httpd/conf/ssl.crt/webdav.mysite.com.crt
    SSLCertificateKeyFile /etc/httpd/conf/ssl.prm/webdav.mysite.com.pem
```

Further reference please see: http://slacksite.com/apache/certificate.html

4.2.5.2 Password protecting the site with htpasswd

The command htpasswd comes with the apache rpm and is the easiest way to place a password protection on a website. The only item you need to be careful about is to keep your password file outside of the document root.

Create the password file and a user

```
htpasswd -c /var/www/htpasswd user
```

Important! Do not use the –c option once the password file has been created – it will overwrite the original file.

```
htpasswd  /var/www/htpasswd user
```

Add the following in the virtualhost directive to protect the site.

```
    <Location />
      AuthUserFile /var/www/htpasswd
      AuthGroupFile /dev/null
      AuthName ByPassword
      AuthType Basic
      Require valid-user
    </Location>
```

4.2.5.3 Configuring webdav

I am using webdav for webserver administration. Since it is already installed by default setting it up is fairly simple. Webdav is secured in two ways the first is the SSL encryption the second is the authentication via the standard htpasswd file.

I have found out that Macromedia Dreamweaver MX 2004 does not support SSL over webdav. The work around is to save all the files locally and copy paste them to the webdav folder in Windows Explorer. For work within the private network I set up another host which is not running SSL so I can continue to use Dreamweaver.

The following should be in the /etc/httpd/conf/httpd.conf file. It was already set up correctly from installation. Make sure that the webserver has access to the lockdb directory.

```
<IfModule mod_dav_fs.c>
    # Location of the WebDAV lock database.
    DAVLockDB /var/lib/dav/lockdb
    DAVDepthInfinity on
</IfModule>
```

Here is an example of the virtualhost using webdav, SSL and htpassword.

```
<VirtualHost 192.168.1.2:443>
    ServerAdmin webmaster@mysite.com
    DocumentRoot /var/www/www.mysite.com
    ServerName webdav.mysite.com
    ErrorLog logs/webdav.mysite.com-error_log
    CustomLog logs/webdav.mysite.com-access_log combined
    SSLEngine on
    SSLCertificateFile /etc/httpd/conf/ssl.crt/webdav.mysite.com.crt
    SSLCertificateKeyFile /etc/httpd/conf/ssl.prm/webdav.mysite.com.pem
    <Location />
```

```
      DAV On
      ForceType text/plain
      DAVMinTimeout 600
      AuthUserFile /var/www/htpasswd
      AuthGroupFile /dev/null
      AuthName ByPassword
      AuthType Basic
      Require valid-user
    </Location>
</VirtualHost>
```

4.2.6 Installing nessus

I want to be able to use nessus on the server. Since nessus can take a long time to complete a run I wanted to be able to run it from the vnc session. That is why I needed to get the gtk+ support enabled for the nessus installation.

Prerequisites:

```
apt-get install XFree86-devel
```

```
wget ftp://ftp.gimp.org/pub/gtk/v1.2/glib-1.2.10.tar.gz
tar zxf glib-1.2.10.tar.gz
cd glib-1.2.10
./configure
make
make install
```

```
wget ftp://ftp.gimp.org/pub/gtk/v1.2/gtk+-1.2.10.tar.gz
cd gtk+-1.2.10
make
make install
```

```
apt-get install sharutils
```

```
wget http://ftp.nessus.org/nessus/nessus-2.0.10a/nessus-installer/nessus-
installer.sh
sh nessus-installer.sh
```

This is the output of the nessus installer

```
Where do you want the whole Nessus package to be installed ?
[/usr/local]
/usr/local/lib is not in /etc/ld.so.conf - shall I add it ? [y]
------------------------------------------------------------------------
-------
                     Nessus installation : Finished
------------------------------------------------------------------------
-------
```

```
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
```

Follow the commands listed after the installation is completed and you are ready to use nessus.

4.2.7 Installing Webmin

I believe Webmin is a good product which can in case of emergency is really useful to have installed. For instance Webmin offers shell access to the server over the web interface. This feature was useful in my past where I accidentally configured the SSH daemon incorrectly and could got gain access to the shell after I logged out of ssh.

In addition to this backdoor feature, Webmin has many modules to configure the various services on the server. In many cases it is easier to use the interface than to search the internet to find the proper functions and syntax of many configuration files. For instance I like using the Webmin for installing additional perl modules on the server.

Because of the large security risk that such a tool can pose on the server the interface will not be forwarded through the router. So access to the interface will only be accessible via the private network and only under the SSL configuration to prevent the root password from being sniffed.

Installation of webmin

```
wget http://aleron.dl.sourceforge.net/sourceforge/webadmin/webmin-1.130-
1.noarch.rpm
```

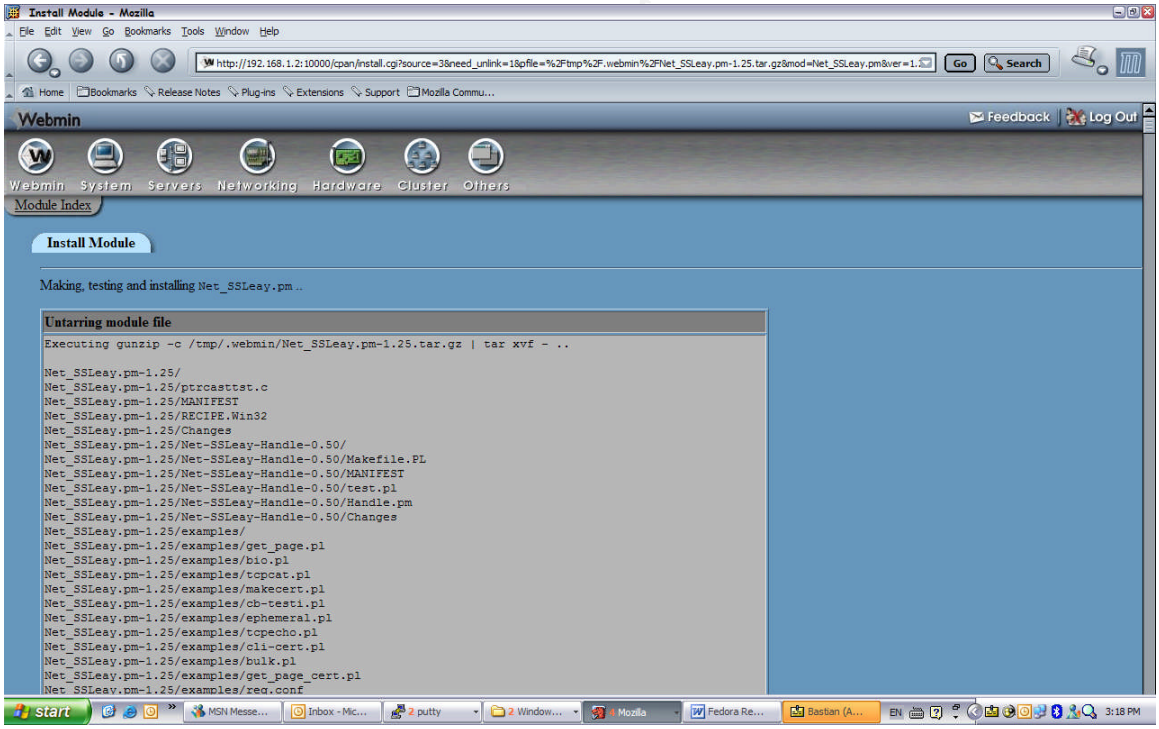
```
rpm --checksig webmin-1.130-1.noarch.rpm
webmin-1.130-1.noarch.rpm: md5 gpg OK

rpm -Uvh webmin-1.130-1.noarch.rpm
```

now we need to install SSL for webmin

```
apt-get install openssl-devel
```

Log onto the Webmin interface (http://server_name.com:10000) log in with the root username and password. Thereafter *click on others→ Perl Modules.*

In the from CPAN select the module named Net::SSLeay then click install. Therafter select *Make test and install* then *Continue with install*.



After this set the server should be accessable via https://server_name.com:10000.

5 Design/Implement Ongoing Maintenance Procedures

Maintaining the server is an ongoing process. There are two main items that are of the biggest concern. The fist maintaining regular backups, the second is making sure the system is up-to-date with the latest security patches.

The server is easy to make backups of. On a monthly basis I will burn a CD with the configuration files and the document root as well as any personal e-mails on the server. There is no tape drive connected to the server so there is no need to use the tool dump to create the backups. Rather, I will create tar balls with incremental on a regular basis then throw them onto the CD. Another option with will come in the future is to use rsync so synchronize critical files on a remote server. This is even better because it allows for the configuration of a hot spare machine that can be up and running as soon as the DNS is updated.

Logwatch is installed by default. It sends a summary of the log files to the root user every day. Taking a quick look at these e-mails gives a good impression if something abnormal is happening on the server.

The other issue is keeping the system up-to-date. For one the administrator needs to be on top of things and read the latest security news. Red Hat maintains a website where it posts the latest security news (http://lwn.net/Alerts/Fedora/). There are a lot of other interesting sites such as http://www.securityfocus.com and the various CERT sites.

5.1 System update

Another item that was set up to ensure the system has the latest versions of software was to place apt-get into the crontab. The script can be found in the previous section of this document. With this the server will automatically download and install any updates that are available for the system.

5.2 System Backup

On a regular basis (monthly) I will burn a CD of critical files. The following directories should be backed up. I am using X-Roast to burn the software via the VNC interface.

- /etc/
- /var/www/
- /var/named/
- /home/

6 Check Configuration

6.1 NMAP - Port scan

The router is set up to only forward certain ports. To double check that only the ports that I specified are accessible from the internet I ran nmap from a foreign host.

TCP connect() port scan (-sT)

```
nmap -sT -P0 xxx.xxx.xxx.xxx
(The 1532 ports scanned but not shown below are in state: filtered)
Port       State       Service
22/tcp     open        ssh
25/tcp     open        smtp
53/tcp     open        domain
80/tcp     open        http
110/tcp    open        pop-3
143/tcp    open        imap2
443/tcp    open        https
465/tcp    closed      smtps
993/tcp    open        imaps
995/tcp    open        pop3s
```

6.2 Run Nessus to check for any security vulnerabilities

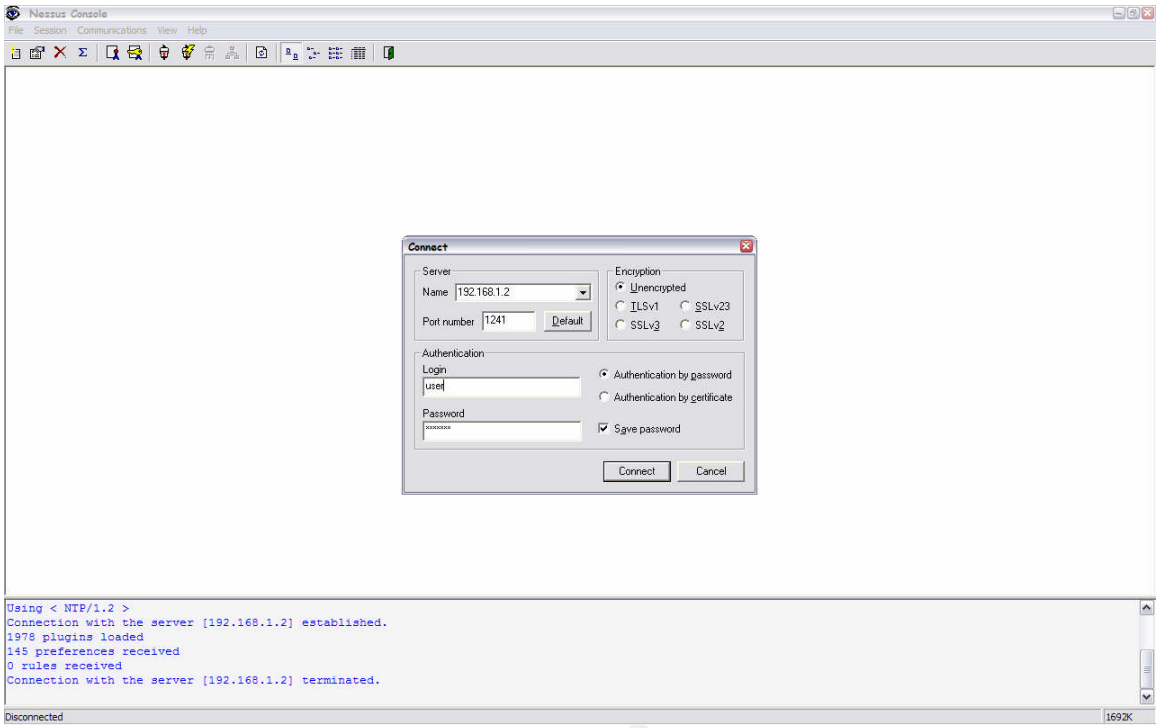
I like to run the nessus client from Windows. Download and install the tool.
http://nessuswx.nessus.org/archive/nessuswx-1.4.4-install.exe

Start the nessus daemon by running the following command:

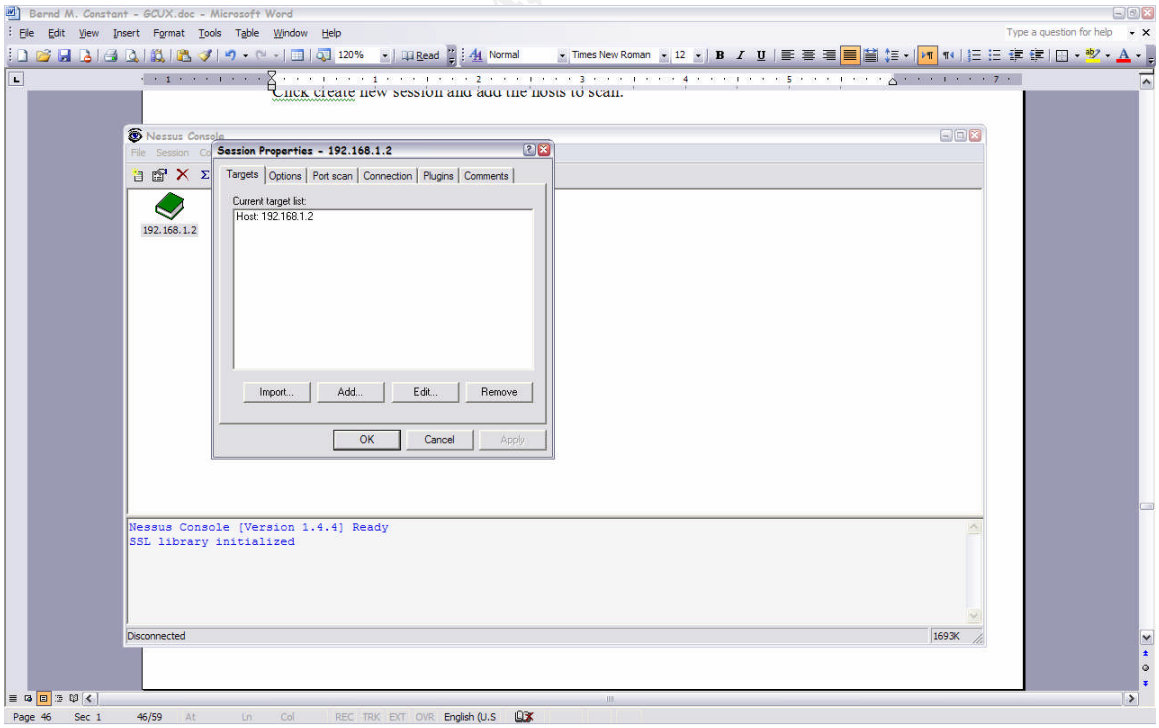
```
[root@hush etc]# nessusd
```

Once you are finished with the scan be sure to turn nessusd back off to avoid a security risk.

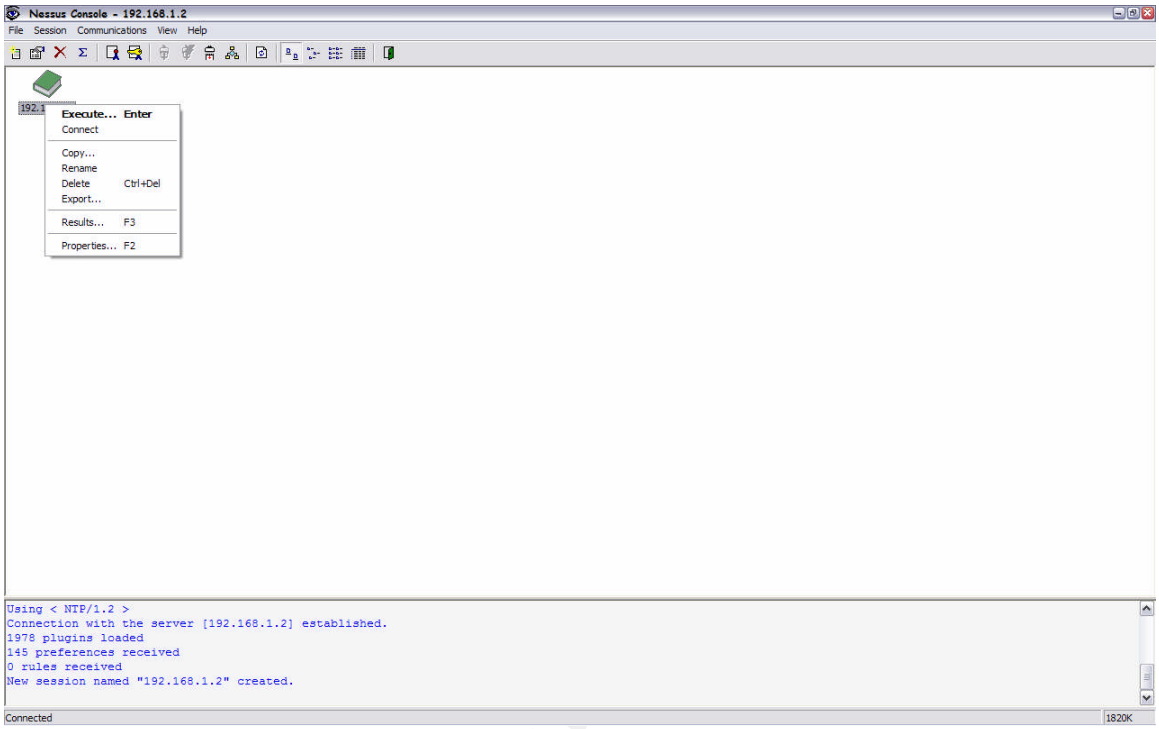
Start Nessus WX and click the connect icon thereafter type in the server ip-address and the user password that you created.



Click create new session and add the hosts to scan.



This creates an icon for the scan. Just select it to start the scan



See the scan results in the appendix.

The scan detected the services that were supposed to run on the server. Since the system was up to date there were no security holes found on the server. The only suggestion was a small change in the apache configuration file. I followed the instructions that Nessus provided me with and solved the issue.

6.3 Test Login

To make sure that the user configuration changes were done correctly I checked that the following users could not log into the machine.

User	Login
Root	From consol
Root	Via ssh
Users not in privileged groups	Via ssh
Loggin on from consol	This should display a banner (configured from bastille)

6.4 Verify system processes

```
# netstat -ap
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
PID/Program name
tcp   0      0 *:imaps                 *:*  LISTEN  3112/xinetd
tcp   0      0 *:pop3s                 *:*  LISTEN  3112/xinetd
tcp   0      0 localhost.localdo:mysql *:*  LISTEN  5926/mysqld
tcp   0      0 *:pop3                  *:*  LISTEN  3112/xinetd
tcp   0      0 localhost.localdoma:783 *:*  LISTEN  3249/spamd -d -c -a
tcp   0      0 *:imap                  *:*  LISTEN  3112/xinetd
tcp   0      0 *:10000                 *:*  LISTEN  3351/perl
tcp   0      0 *:http                  *:*  LISTEN  3261/httpd
tcp   0      0 hush:domain             *:*  LISTEN  3082/named
tcp   0      0 localhost.locald:domain *:*  LISTEN  3082/named
tcp   0      0 *:ssh                   *:*  LISTEN  3096/sshd
tcp   0      0 *:smtp                  *:*  LISTEN  3228/sendmail: acce
tcp   0      0 localhost.localdom:rndc *:*  LISTEN  3082/named
tcp   0      0 *:https                 *:*  LISTEN  3261/httpd
udp   0      0 *:10000                 *:*  3351/perl
udp   0      0 *:domain                *:*  3082/named
udp   0      0 hush:domain             *:*  3082/named
udp   0      0 localhost.locald:domain *:*  3082/named
udp   0      0 hush:ntp                *:*  3126/ntpd
udp   0      0 localhost.localdoma:ntp *:*  3126/ntpd
udp   0      0 *:ntp                   *:*  3126/ntpd
```

This confirms that only the services that I intended to install are listening on the ports. Webmin is running on port 10000. It is only accessible via the private network since it is not forwarded through the router.

- Spamd is used for sendmail and bound to localhost.
- MySQL is bound to localhost.

6.5 Check that the chattr attribute works

In order to check if the chattr attribute works. Try to edit the file /etc/hosts. It should not allow you to edit the file.

7 Appendix

7.1 Partitioning

This is the only part of the server installation that I am not 100% happy about. When I set up the server I could not decide on partition sizes. Now that I have the server installed I would put almost all the space on /var and /home that way I can install quotas more easy.

Total HD space 112580MB

Device	Mount Point	Type	Format	Size	Start	End
/dev/hda1	/boot	/ext3	Y	102	1	13
/dev/hda2	/	/ext3	Y	112581	14	14365
/dev/hda3		swap	Y	1788	14366	14593

7.2 Bastille configuration file

```
cat /etc/Bastille/config
# Q: Would you like to set more restrictive permissions on the
administration utilities? [N]
FilePermissions.generalperms_1_1="Y"
# Q: Would you like to disable SUID status for mount/umount?
FilePermissions.suidmount="Y"
# Q: Would you like to disable SUID status for ping? [Y]
FilePermissions.suidping="Y"
# Q: Would you like to disable SUID status for at? [Y]
FilePermissions.suidat="Y"
# Q: Would you like to disable SUID status for usernetctl? [Y]
FilePermissions.suidusernetctl="Y"
# Q: Would you like to disable SUID status for traceroute? [Y]
FilePermissions.suidtrace="Y"
# Q: Would you like to disable SUID status for XFree86? [N]
FilePermissions.suidXFree86="N"
# Q: Would you like to enforce password aging? [Y]
AccountSecurity.passwdage="N"
# Q: Would you like to restrict the use of cron to administrative
accounts? [Y]
AccountSecurity.cronuser="Y"
# Q: Do you want to set the default umask? [Y]
AccountSecurity.umaskyn="Y"
# Q: What umask would you like to set for users on the system? [077]
AccountSecurity.umask="077"
# Q: Should we disallow root login on tty's 1-6? [N]
AccountSecurity.rootttylogins="Y"
# Q: Would you like to password-protect the GRUB prompt? [N]
BootSecurity.protectgrub="Y"
# Q: Enter GRUB password, please.   []
```

```
BootSecurity.protectgrub_password="letme1n"
# Q: Would you like to disable CTRL-ALT-DELETE rebooting? [N]
BootSecurity.secureinittab="Y"
# Q: Would you like to password protect single-user mode? [Y]
BootSecurity.passsum="Y"
# Q: Would you like to set a default-deny on TCP Wrappers and xinetd? [N]
SecureInetd.tcpd_default_deny="Y"
# Q: Should Bastille ensure the telnet service does not run on this
system? [y]
SecureInetd.deactivate_telnet="Y"
# Q: Should Bastille ensure inetd's FTP service does not run on this
system? [y]
SecureInetd.deactivate_ftp="Y"
# Q: Would you like to display "Authorized Use" messages at log-in time?
[Y]
SecureInetd.banners="Y"
# Q: Who is responsible for granting authorization to use this machine?
SecureInetd.owner="BerndMatt"
# Q: Would you like to disable the gcc compiler? [N]
DisableUserTools.compiler="N"
# Q: Would you like to put limits on system resource usage? [N]
ConfigureMiscPAM.limitsconf="N"
# Q: Should we restrict console access to a small group of user accounts?
[N]
ConfigureMiscPAM.consolelogin="N"
# Q: Would you like to add additional logging? [Y]
Logging.morelogging="Y"
# Q: Do you have a remote logging host? [N]
Logging.remotelog="N"
# Q: Would you like to disable apmd? [Y]
MiscellaneousDaemons.apmd="Y"
# Q: Do you want to stop sendmail from running in daemon mode? [Y]
Sendmail.sendmaildaemon="N"
# Q: Would you like to deactivate named, at least for now? [Y]
DNS.namedoff="N"
# Q: Would you like to deactivate the Apache web server? [Y]
Apache.apacheoff="N"
# Q: Would you like to bind the web server to listen only to the
localhost? [N]
Apache.bindapachelocal="N"
# Q: Would you like to bind the web server to a particular interface? [N]
Apache.bindapachenic="N"
# Q: Would you like to deactivate the following of symbolic links? [Y]
Apache.symlink="N"
# Q: Would you like to deactivate server-side includes? [Y]
Apache.ssi="N"
# Q: Would you like to disable CGI scripts, at least for now? [Y]
Apache.cgi="N"
# Q: Would you like to disable indexes? [N]
Apache.apacheindex="N"
# Q: Would you like to install TMPDIR/TMP scripts? [N]
TMPDIR.tmpdir="N"
# Q: Would you like to run the packet filtering script? [N]
Firewall.ip_intro="N"
```

7.3 Nessus scan results

```
192.168.1.2  ssh (22/tcp)
192.168.1.2  smtp (25/tcp)
192.168.1.2  domain (53/tcp)
192.168.1.2  http (80/tcp)
192.168.1.2  pop3 (110/tcp)
192.168.1.2  imap (143/tcp)
192.168.1.2  https (443/tcp)
192.168.1.2  imaps (993/tcp)
192.168.1.2  pop3s (995/tcp)
192.168.1.2  http (80/tcp)10330  INFO   "A web server is running on this
port
"
192.168.1.2  domain (53/udp)     11002  INFO   "
A DNS server is running on this port. If you do not use it, disable it.

Risk factor : Low
"
192.168.1.2  domain (53/udp)
192.168.1.2  domain (53/tcp)     11002  INFO   "
A DNS server is running on this port. If you do not use it, disable it.

Risk factor : Low
"
192.168.1.2  ssh (22/tcp) 10330  INFO   "An ssh server is running on this
port
"
192.168.1.2  smtp (25/tcp)10330  INFO   "An SMTP server is running on
this port
Here is its banner :
220 192.168.1.2.spliffnet.com ESMTP Sendmail 8.12.10/8.12.10
 Sun, 23 May 2004 17:22:16 -0400

"
192.168.1.2  pop3 (110/tcp)      10330  INFO   "A pop3 server is running
on this port
"
192.168.1.2  https (443/tcp)     10330  INFO   "An unknown service is
running on this port.
It is usually reserved for HTTPS
"
192.168.1.2  pop3s (995/tcp)     10330  INFO   "An unknown service is
running on this port.
It is usually reserved for POP3S
"
192.168.1.2  domain (53/tcp)     10028  INFO   "BIND 'NAMED' is an open-
source DNS server from ISC.org.
Many proprietary DNS servers are based on BIND source code.

The BIND based NAMED servers (or DNS servers) allow remote users
to query for version and type information.  The query of the CHAOS
TXT record 'version.bind', will typically prompt the server to send
the information back to the querying source.

The remote bind version is : 9.2.2-P3

Solution :
Using the 'version' directive in the 'options' section will block
the 'version.bind' query, but it will not log such attempts.
```

```
"
192.168.1.2  imap (143/tcp)      10330  INFO   "An IMAP server is running
on this port
"
192.168.1.2  domain (53/tcp)     10539  INFO   "
The remote name server allows recursive queries to be performed
by the host running nessusd.

If this is your internal nameserver, then forget this warning.

If you are probing a remote nameserver, then it allows anyone
to use it to resolve third parties names (such as www.nessus.org).
This allows hackers to do cache poisoning attacks against this
nameserver.

If the host allows these recursive queries via UDP,
then the host can be used to 'bounce' Denial of Service attacks
against another network or system.

See also : http://www.cert.org/advisories/CA-1997-22.html

Solution : Restrict recursive queries to the hosts that should
use this nameserver (such as those of the LAN connected to it).

If you are using bind 8, you can do this by using the instruction
'allow-recursion' in the 'options' section of your named.conf

If you are using bind 9, you can define a grouping of internal addresses
using the 'acl' command

Then, within the options block, you can explicitly state:
'allow-recursion { hosts_defined_in_acl }'

For more info on Bind 9 administration (to include recursion), see:
http://www.nominum.com/content/documents/bind9arm.pdf

If you are using another name server, consult its documentation.

Risk factor : Serious
CVE : CVE-1999-0024
BID : 678
"
192.168.1.2  imaps (993/tcp)     10330  INFO   "An unknown service is
running on this port.
It is usually reserved for IMAPS
"
192.168.1.2  smtp (25/tcp)10263  INFO   "Remote SMTP server banner :
220 192.168.1.2.spliffnet.com ESMTP Sendmail 8.12.10/8.12.10
 Sun, 23 May 2004 17:22:26 -0400




This is probably: Sendmail version 8.12.10


"
192.168.1.2  ssh (22/tcp) 10267  INFO   "Remote SSH version : SSH-2.0-
OpenSSH_3.6.1p2
"
192.168.1.2  domain (53/udp)     11951  INFO   "The remote name server
could be fingerprinted as being one of the following :
ISC BIND 9.2.1
```

```
ISC BIND 9.2.2

"
192.168.1.2  ssh (22/tcp) 10881  INFO   "The remote SSH daemon supports
the following versions of the
SSH protocol :

  . 1.99
  . 2.0

"
192.168.1.2  http (80/tcp)11032  INFO   "The following directories were
discovered:
/cgi-bin, /error, /icons, /mailman, /manual

While this is not, in and of itself, a bug, you should manually inspect
these directories to ensure that they are in compliance with company
security standards

"
192.168.1.2  http (80/tcp)11213  INFO   "
Your webserver supports the TRACE and/or TRACK methods. TRACE and TRACK
are HTTP methods which are used to debug web server connections.

It has been shown that servers supporting this method are subject
to cross-site-scripting attacks, dubbed XST for
""Cross-Site-Tracing"", when used in conjunction with
various weaknesses in browsers.

An attacker may use this flaw to trick your
legitimate web users to give him their
credentials.

Solution: Disable these methods.


If you are using Apache, add the following lines for each virtual
host in your configuration file :

    RewriteEngine on
    RewriteCond %{REQUEST_METHOD} ^(TRACE|TRACK)
    RewriteRule .* - [F]

If you are using Microsoft IIS, use the URLScan tool to deny HTTP TRACE
requests or to permit only the methods needed to meet site requirements
and policy.

If you are using Sun ONE Web Server releases 6.0 SP2 and later, add the
following to the default object section in obj.conf:
    <Client method=""TRACE"">
     AuthTrans fn=""set-variable""
     remove-headers=""transfer-encoding""
     set-headers=""content-length: -1""
     error=""501""
    </Client>

If you are using Sun ONE Web Server releases 6.0 SP2 or below, compile
the NSAPI plugin located at:
   http://sunsolve.sun.com/pub-cgi/retrieve.pl?doc=fsalert%2F50603


See http://www.whitehatsec.com/press_releases/WH-PR-20030120.pdf
    http://archives.neohapsis.com/archives/vulnwatch/2003-q1/0035.html
```

```
    http://sunsolve.sun.com/pub-cgi/retrieve.pl?doc=fsalert%2F50603
    http://www.kb.cert.org/vuls/id/867593

Risk factor : Medium
"
192.168.1.2  http (80/tcp) 11919  INFO   "This web server was
fingerprinted as: Apache/2.0.4x on Redhat Linux 9 / Fedora
which is not consistent with the displayed banner: Apache

If you think that Nessus was wrong, please send this signature
to www-signatures@nessus.org :
HTM:200:200:200:200:200:HTM:200:200:200:HTM:HTM:200:400:400:400:400:404:4
05:405:200:200:405:405:200:FIXME:Apache
Including these headers:
X-Powered-By: PHP/4.3.4

"
192.168.1.2  http (80/tcp) 10107  INFO   "The remote web server type is :

Apache


and the 'ServerTokens' directive is ProductOnly
Apache does not permit to hide the server type.

"
192.168.1.2  X11:1 (6001/tcp)    10407  INFO   "This X server does *not*
allow any client to connect to it
however it is recommended that you filter incoming connections
to this port as attacker may send garbage data and slow down
your X session or even kill the server.

Here is the server version : 11.0
Here is the message we received : No protocol specified


Solution : filter incoming connections to ports 6000-6009
Risk factor : Low
CVE : CVE-1999-0526
"
192.168.1.2  X11:1 (6001/tcp)
192.168.1.2  ssh (22/tcp) 11837  REPORT "
You are running a version of OpenSSH which is older than 3.7.1

Versions older than 3.7.1 are vulnerable to a flaw in the buffer
management
functions which might allow an attacker to execute arbitrary commands on
this
host.

An exploit for this issue is rumored to exist.


Note that several distribution patched this hole without changing
the version number of OpenSSH. Since Nessus solely relied on the
banner of the remote SSH server to perform this check, this might
be a false positive.

If you are running a RedHat host, make sure that the command :
          rpm -q openssh-server

Returns :
 openssh-server-3.1p1-13 (RedHat 7.x)
```

```
 openssh-server-3.4p1-7  (RedHat 8.0)
 openssh-server-3.5p1-11 (RedHat 9)

Solution : Upgrade to OpenSSH 3.7.1
See also : http://marc.theaimsgroup.com/?l=openbsd-
misc&m=106375452423794&w=2
    http://marc.theaimsgroup.com/?l=openbsd-misc&m=106375456923804&w=2
Risk factor : High
CVE : CAN-2003-0682, CAN-2003-0693, CAN-2003-0695
BID : 8628
Other references : RHSA:RHSA-2003:279-02, SuSE:SUSE-SA:2003:039
"
192.168.1.2  imap (143/tcp)      11414  INFO   "The remote imap server
banner is :
* OK [CAPABILITY IMAP4REV1 LOGIN-REFERRALS STARTTLS AUTH=LOGIN]
192.168.1.2 IMAP4rev1 2003.338rh at Sun, 23 May 2004 17:22:18 -0400 (EDT)

Versions and types should be omitted where possible.
Change the imap banner to something generic.
"
192.168.1.2  vnc-1 (5901/tcp)    10342  INFO   "
The remote server is running VNC.
VNC permits a console to be displayed remotely.

Solution: Disable VNC access from the network by
using a firewall, or stop VNC service if not needed.

Risk factor : Medium
"
192.168.1.2  vnc-1 (5901/tcp)
192.168.1.2  vnc-1 (5901/tcp)    10342  INFO   "Version of VNC Protocol
is: RFB 003.007

"
192.168.1.2  pop3 (110/tcp)      10185  INFO   "
The remote POP3 servers leak information about the software it is
running,
through the login banner. This may assist an attacker in choosing an
attack
strategy.

Versions and types should be omitted where possible.

The version of the remote POP3 server is :
+OK  192.168.1.2 v2003.83rh server ready


Solution : Change the login banner to something generic.
Risk factor : Low
"
192.168.1.2  https (443/tcp)     11154  INFO   "An unknown server is
running on this port.
If you know what it is, please send this banner to the Nessus team:
00: 3c 3f 78 6d 6c 20 76 65 72 73 69 6f 6e 3d 22 31    <?xml version=""1
10: 2e 30 22 20 65 6e 63 6f 64 69 6e 67 3d 22 49 53    .0"" encoding=""IS
20: 4f 2d 38 38 35 39 2d 31 22 3f 3e 0a 3c 21 44 4f    O-8859-1""?>.<!DO
30: 43 54 59 50 45 20 68 74 6d 6c 20 50 55 42 4c 49    CTYPE html PUBLI
40: 43 20 22 2d 2f 2f 57 33 43 2f 2f 44 54 44 20 58    C ""-//W3C//DTD X
50: 48 54 4d 4c 20 31 2e 30 20 53 74 72 69 63 74 2f    HTML 1.0 Strict/
60: 2f 45 4e 22 0a 20 20 22 68 74 74 70 3a 2f 2f 77    /EN"".  ""http://w
70: 77 77 2e 77 33 2e 6f 72 67 2f 54 52 2f 78 68 74    ww.w3.org/TR/xht
80: 6d 6c 31 2f 44 54 44 2f 78 68 74 6d 6c 31 2d 73    ml1/DTD/xhtml1-s
90: 74 72 69 63 74 2e 64 74 64 22 3e 0a 3c 68 74 6d    trict.dtd"">.<htm
```

```
a0: 6c 20 78 6d 6c 6e 73 3d 22 68 74 74 70 3a 2f 2f    l xmlns=""http://
b0: 77 77 77 2e 77 33 2e 6f 72 67 2f 31 39 39 39 2f    www.w3.org/1999/
c0: 78 68 74 6d 6c 22 20 6c 61 6e 67 3d 22 65 6e 22    xhtml""
lang=""en""
d0: 20 78 6d 6c 3a 6c 61 6e 67 3d 22 65 6e 22 3e 0a     xml:lang=""en"">.
e0: 3c 68 65 61 64 3e 0a 3c 74 69 74 6c 65 3e 42 61    <head>.<title>Ba
f0: 64 20 72 65 71 75 65 73 74 21 3c 2f 74 69 74 6c    d request!</titl
00: 65 3e 0a 3c 6c 69 6e 6b 20 72 65 76 3d 22 6d 61    e>.<link rev=""ma
10: 64 65 22 20 68 72 65 66 3d 22 6d 61 69 6c 74 6f    de"" href=""mailto
20: 3a 77 65 62 6d 61 73 74 65 72 40 73 70 6c 69 66    :webmaster@splif
30: 66 6e 65 74 2e 63 6f 6d 22 20 2f 3e 0a 3c 73 74    fnet.com"" />.<st
40: 79 6c 65 20 74 79 70 65 3d 22 74 65 78 74 2f 63    yle type=""text/c
50: 73 73 22 3e 3c 21 2d 2d 2f 2a 2d 2d 3e 3c 21 5b    ss""><!--/*--><![
60: 43 44 41 54 41 5b 2f 2a 3e 3c 21 2d 2d 2a 2f 20    CDATA[/*><!--*/
70: 0a 20 20 20 20 62 6f 64 79 20 7b 20 63 6f 6c 6f    .    body { colo
80: 72 3a 20 23 30 30 30 30 30 30 3b 20 62 61 63 6b    r: #000000
 back
90: 67 72 6f 75 6e 64 2d 63 6f 6c 6f 72 3a 20 23 46    ground-color: #F
a0: 46 46 46 46 46 3b 20 7d 0a 20 20 20 20 61 3a 6c    FFFFF
 }.    a:l
b0: 69 6e 6b 20 7b 20 63 6f 6c 6f 72 3a 20 23 30 30    ink { color: #00
c0: 30 30 43 43 3b 20 7d 0a 20 20 20 20 70 2c 20 61    00CC
 }.    p, a
d0: 64 64 72 65 73 73 20 7b 6d 61 72 67 69 6e 2d 6c    ddress {margin-l
e0: 65 66 74 3a 20 33 65 6d 3b 7d 0a 20 20 20 20 73    eft: 3em
}.    s
f0: 70 61 6e 20 7b 66 6f 6e 74 2d 73 69 7a 65 3a 20    pan {font-size:
00: 73 6d 61 6c 6c 65 72 3b 7d 0a 2f 2a 5d 5d 3e 2a    smaller
}./*]]>*
10: 2f 2d 2d 3e 3c 2f 73 74 79 6c 65 3e 0a 3c 2f 68    /--></style>.</h
20: 65 61 64 3e 0a 0a 3c 62 6f 64 79 3e 0a 3c 68 31    ead>..<body>.<h1
30: 3e 42 61 64 20 72 65 71 75 65 73 74 21 3c 2f 68    >Bad request!</h
40: 31 3e 0a 3c 70 3e 0a 0a 20 20 20 20 59 6f 75 72    1>.<p>..    Your
50: 20 62 72 6f 77 73 65 72 20 28 6f 72 20 70 72 6f     browser (or pro
60: 78 79 29 20 73 65 6e 74 20 61 20 72 65 71 75 65    xy) sent a reque
70: 73 74 20 74 68 61 74 0a 20 20 20 20 74 68 69 73    st that.    this
80: 20 73 65 72 76 65 72 20 63 6f 75 6c 64 20 6e 6f     server could no
90: 74 20 75 6e 64 65 72 73 74 61 6e 64 2e 0a 0a 3c    t understand...<
a0: 2f 70 3e 0a 3c 70 3e 0a 49 66 20 79 6f 75 20 74    /p>.<p>.If you t
b0: 68 69 6e 6b 20 74 68 69 73 20 69 73 20 61 20 73    hink this is a s
c0: 65 72 76 65 72 20 65 72 72 6f 72 2c 20 70 6c 65    erver error, ple
d0: 61 73 65 20 63 6f 6e 74 61 63 74 0a 74 68 65 20    ase contact.the
e0: 3c 61 20 68 72 65 66 3d 22 6d 61 69 6c 74 6f 3a    <a href=""mailto:
f0: 77 65 62 6d 61 73 74 65 72 40 73 70 6c 69 66 66    webmaster@spliff
00: 6e 65 74 2e 63 6f 6d 22 3e 77 65 62 6d 61 73 74    net.com"">webmast
10: 65 72 3c 2f 61 3e 2e 0a 0a 3c 2f 70 3e 0a 0a 3c    er</a>...</p>..<
20: 68 32 3e 45 72 72 6f 72 20 34 30 30 3c 2f 68 32    h2>Error 400</h2
30: 3e 0a 3c 61 64 64 72 65 73 73 3e 0a 20 20 3c 61    >.<address>.  <a
40: 20 68 72 65 66 3d 22 2f 22 3e 6d 61 69 6c 2e 63     href=""/"">mail.c
50: 6f 6e 73 74 61 6e 74 2e 61 74 3c 2f 61 3e 3c 62    onstant.at</a><b
60: 72 20 2f 3e 0a 20 20 3c 73 70 61 6e 3e 41 70 61    r />.  <span>Apa
70: 63 68 65 3c 2f 73 70 61 6e 3e 0a 3c 2f 61 64 64    che</span>.</add
80: 72 65 73 73 3e 0a 3c 2f 62 6f 64 79 3e 0a 3c 2f    ress>.</body>.</
90: 68 74 6d 6c 3e 0a 0a                               html>..


"
```

7.4 Sendmail.conf

```
divert(-1)dnl
dnl #
dnl # This is the sendmail macro config file for m4. If you make changes
to
dnl # /etc/mail/sendmail.mc, you will need to regenerate the
dnl # /etc/mail/sendmail.cf file by confirming that the sendmail-cf
package is
dnl # installed and then performing a
dnl #
dnl #     make -C /etc/mail
dnl #
include(`/usr/share/sendmail-cf/m4/cf.m4')dnl
VERSIONID(`setup for Red Hat Linux')dnl
OSTYPE(`linux')dnl
dnl #
dnl # Uncomment and edit the following line if your outgoing mail needs
to
dnl # be sent out through an external mail server:
dnl #
define(`SMART_HOST',`xxx.xxx.xxx.xxx')
dnl #
define(`confDEF_USER_ID',``8:12'')dnl
dnl define(`confAUTO_REBUILD')dnl
define(`confLOG_LEVEL', `10')dnl
define(`confTO_CONNECT', `1m')dnl
define(`confTRY_NULL_MX_LIST',true)dnl
define(`confDONT_PROBE_INTERFACES',true)dnl
define(`PROCMAIL_MAILER_PATH',`/usr/bin/procmail')dnl
define(`ALIAS_FILE', `/etc/aliases')dnl
define(`STATUS_FILE', `/etc/mail/statistics')dnl
define(`UUCP_MAILER_MAX', `2000000')dnl
define(`confUSERDB_SPEC', `/etc/mail/userdb.db')dnl
define(`confPRIVACY_FLAGS', `authwarnings,novrfy,noexpn,restrictqrun')dnl
define(`confAUTH_OPTIONS', `A')dnl
dnl #
dnl # The following allows relaying if the user authenticates, and
disallows
dnl # plaintext authentication (PLAIN/LOGIN) on non-TLS links
dnl #
dnl define(`confAUTH_OPTIONS', `A p')dnl
dnl #
dnl # PLAIN is the preferred plaintext authentication method and used by
dnl # Mozilla Mail and Evolution, though Outlook Express and other MUAs
do
dnl # use LOGIN. Other mechanisms should be used if the connection is not
dnl # guaranteed secure.
dnl #
TRUST_AUTH_MECH(`EXTERNAL DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dnl
define(`confAUTH_MECHANISMS', `EXTERNAL GSSAPI DIGEST-MD5 CRAM-MD5 LOGIN
PLAIN')
dnl
dnl #
dnl # Rudimentary information on creating certificates for sendmail TLS:
dnl #     make -C /usr/share/ssl/certs usage
dnl #
 define(`confCACERT_PATH',`/usr/share/ssl/certs')
 define(`confCACERT',`/usr/share/ssl/certs/ca-bundle.crt')
```

```
 define(`confSERVER_CERT',`/usr/share/ssl/certs/sendmail.pem')
 define(`confSERVER_KEY',`/usr/share/ssl/certs/sendmail.pem')
dnl #
dnl # This allows sendmail to use a keyfile that is shared with
OpenLDAP's
dnl # slapd, which requires the file to be readble by group ldap
dnl #
dnl define(`confDONT_BLAME_SENDMAIL',`groupreadablekeyfile')dnl
dnl #
dnl define(`confTO_QUEUEWARN', `4h')dnl
dnl define(`confTO_QUEUERETURN', `5d')dnl
dnl define(`confQUEUE_LA', `12')dnl
dnl define(`confREFUSE_LA', `18')dnl
define(`confTO_IDENT', `0')dnl
dnl FEATURE(delay_checks)dnl
FEATURE(`no_default_msa',`dnl')dnl
FEATURE(`smrsh',`/usr/sbin/smrsh')dnl
FEATURE(`mailertable',`hash -o /etc/mail/mailertable.db')dnl
FEATURE(`virtusertable',`hash -o /etc/mail/virtusertable.db')dnl
FEATURE(redirect)dnl
FEATURE(always_add_domain)dnl
FEATURE(use_cw_file)dnl
FEATURE(use_ct_file)dnl
dnl #
dnl # The -t option will retry delivery if e.g. the user runs over his
quota.
dnl #
FEATURE(local_procmail,`',`procmail -t -Y -a $h -d $u')dnl
FEATURE(`access_db',`hash -T<TMPF> -o /etc/mail/access.db')dnl
FEATURE(`blacklist_recipients')dnl
EXPOSED_USER(`root')dnl
dnl #
dnl # The following causes sendmail to only listen on the IPv4 loopback
address
dnl # 127.0.0.1 and not on any other network devices. Remove the loopback
dnl # address restriction to accept email from the internet or intranet.
dnl #
dnl DAEMON_OPTIONS(`Port=smtp,Addr=127.0.0.1, Name=MTA')dnl
dnl DAEMON_OPTIONS(`Port=smtp, Name=MTA')dnl
dnl #
dnl # The following causes sendmail to additionally listen to port 587
for
dnl # mail from MUAs that authenticate. Roaming users who can't reach
their
dnl # preferred sendmail daemon due to port 25 being blocked or
redirected find
dnl # this useful.
dnl #
dnl DAEMON_OPTIONS(`Port=submission, Name=MSA, M=Ea')dnl
dnl #
dnl # The following causes sendmail to additionally listen to port 465,
but
dnl # starting immediately in TLS mode upon connecting. Port 25 or 587
followed
dnl # by STARTTLS is preferred, but roaming clients using Outlook Express
can't
dnl # do STARTTLS on ports other than 25. Mozilla Mail can ONLY use
STARTTLS
dnl # and doesn't support the deprecated smtps; Evolution <1.1.1 uses
smtps
dnl # when SSL is enabled-- STARTTLS support is available in version
1.1.1.
dnl #
```

```
dnl # For this to work your OpenSSL certificates must be configured.
dnl #
dnl DAEMON_OPTIONS(`Port=smtps, Name=TLSMTA, M=s')dnl
dnl #
dnl # The following causes sendmail to additionally listen on the IPv6
loopback
dnl # device. Remove the loopback address restriction listen to the
network.
dnl #
dnl # NOTE: binding both IPv4 and IPv6 daemon to the same port requires
dnl #       a kernel patch
dnl #
dnl DAEMON_OPTIONS(`port=smtp,Addr=::1, Name=MTA-v6, Family=inet6')dnl
dnl #
dnl # We strongly recommend not accepting unresolvable domains if you
want to
dnl # protect yourself from spam. However, the laptop and users on
computers
dnl # that do not have 24x7 DNS do need this.
dnl #
dnl FEATURE(`accept_unresolvable_domains')dnl
dnl #
FEATURE(`relay_based_on_MX')dnl
dnl #
dnl # Also accept email sent to "localhost.localdomain" as local email.
dnl #
LOCAL_DOMAIN(`localhost.localdomain')dnl
dnl #
dnl # The following example makes mail from this host and any additional
dnl # specified domains appear to be sent from mydomain.com
dnl #
dnl MASQUERADE_AS(`mydomain.com')dnl
dnl #
dnl # masquerade not just the headers, but the envelope as well
dnl #
dnl FEATURE(masquerade_envelope)dnl
dnl #
dnl # masquerade not just @mydomainalias.com, but @*.mydomainalias.com as
well
dnl #
dnl FEATURE(masquerade_entire_domain)dnl
dnl #
dnl MASQUERADE_DOMAIN(localhost)dnl
dnl MASQUERADE_DOMAIN(localhost.localdomain)dnl
dnl MASQUERADE_DOMAIN(mydomainalias.com)dnl
dnl MASQUERADE_DOMAIN(mydomain.lan)dnl
MAILER(smtp)dnl
MAILER(procmail)dnl
```