Building a Secure RedHat Web and FTP Server	
Michael P. Thompson Hilgraeve Inc. GIAC# miket001 April 04, 2001	Track 6 - Unix Security SANS Security 2001 New Orleans GCUX Practical Assignment Version 1.6b

Summary:

The purpose of this document is to provide a checklist that can server as step-by-step guide to building a secure Internet server running Linux. The examples given will focus on RedHat Linux 6.2 running on a rack-mount dual Pentium III server, and will cover everything from operating system installation to installation in data center. When done, we will have a fully-functional secure Internet server, with secure and non-secure web, and anonymous ftp capabilities.

Assumptions:

- Setup will occur on a secure network residing behind a properly configured firewall.
- Servers will be installed in a locked cabinet at a secured colocation facility.
- Local network at colocation facility is protected by firewall blocking all unauthorized traffic.
- No local users, other than administrators.
- Remote syslog, backup, and time services exists on local network in colocation facility.

Pre-Installation	
Determine role of server	
Before building a server, you must determine what the role of that server will be. That way, you only install the services that are needed. Additional services can be added later, if required. For this exercise, we will be building an Internet server running Apache with mod_ssl to provide both secure and non-secure web server capabilities. We will also be using the anonftp package from RedHat to provide anonymous ftp server capabilities. Server administration will be done via ssh. No other services will be required or running on this server.	
Obtain necessary network information	
Assign IP addresses relevant for your network. You may need to contact your Systems Administrator for assistance.	
Initial network: eth0: IP address___.___.___.___ Net Mask___.___.___.___ Broadcast___.___.___.___ Network___.___.___.___ eth1: IP address___.___.___.___ Net Mask___.___.___.___ Broadcast___.___.___.___ Network___.___.___.___ Gateway___.___.___.___	Data center network: eth0: IP address___.___.___.___ Net Mask___.___.___.___ Broadcast___.___.___.___ Network___.___.___.___ eth1: IP address___.___.___.___ Net Mask___.___.___.___ Broadcast___.___.___.___ Network___.___.___.___ Gateway___.___.___.___ Pri. DNS___.___.___.___ Sec. DNS___.___.___.___

Pri. DNS___.___.___.___ Sec. DNS___.___.___.___	
Determine admin accounts and passwords	
Don't rely on using the root account for administration of the server. Assign a username and password for each administrator of the server. You should have at least a primary and secondary administrator. Use good passwords. Use a random password generator to assign passwords that are a combination of numbers, symbols, and characters of mixed case.	
Primary Admin: Name_____ Username_____ Password_____ Phone_____ Email_____ Pager_____	Secondary Admin: Name_____ Username_____ Password_____ Phone_____ Email_____ Pager_____
Record hardware configuration	
Good asset management requires knowing what hardware you own or control. There are various other reasons for keeping good books on hardware information, including several security related items. A record of this information is very useful when the server is mounted in a locked cabinet in a data center located hours away.	
Mfgr._____ Model_____ Serial_____ Service tag_____	CPU: Type_____Mhz_____No._____ RAM_____ HDD: Size_____No.____RAID____ NIC_____No.____ MAC Address_____ Hotswap: Po.Supply____HDD____
Record Support Information	
Record the vendor support information. You never know when you'll need it. And by the time yo need it, you don't want to spend a lot of time tracking this information down.	
Vendor_____ Phone_____ Email_____ Warranty expiration_____	Priority id_____ Username_____ Password_____
Apply for server certificate from a Certificate Authority	
Server certificates must be obtained by a trusted root Certificate Authority (CA). 128-bit, or high encryption, certificates will be used by the web server to authenticate with and encrypt traffic from users. When selecting a root CA, consider the target audience of the web site in that older version of browsers do not include all of the latest CA's.	
Steps: 1. Generate server key. 2. Generate certificate request. 3. Copy key and request to floppy. 4. Apply for server certificate. 5. Pick up certificate. 6. Make backup of floppy and store	# cd /etc/httpd/conf # make genkey Using a blank passphrase trades security for the ability to bring up the web server unattended. Record passphrase if one is used _____ # make certreq Country Name: US State or Province: Michigan Locality...: Monroe Company ...: Hilgraeve

	Department...: Server Host Name: *hostname.domain.com* Admin EMail: admin@domain.com challenge password: _____ optional company name: # cp ssl.csr/server.csr /mnt/floppy # cp ssl.key/server.key /mnt/floppy Using a web browser, go to Verisign's or Thawte's web site and fill out online request forms. Record request password, if used_____ When the certificate has been issued, the issuer will send you email with instruction to pick up your certificate. Save the certificate to floppy. If using Verisign, you should also pick up the Verisign global site id available on their web site. Save it on the floppy as gsid.crt. Keep server keys and certificates secure. They can be used by hackers to impersonate you or your company.

Prepare installation materials
The most secure way to get components onto a server while building it is to put the components onto a CD beforehand. We will be installing components from two CD's -- RedHat 6.2, and a custom CD containing the additional components and updates we wish to install. Server components that are available in source code form only have been compiled on another computer and packaged into an RPM file. We will also need the floppy containing the server key and certificate that we created in the previous step, as well as an additional floppy that will be used as a boot disk for the system.

Server Installation and Configuration	
Create boot disk for RedHat 6.2	
From a Windows or DOS machine with CD access.	Insert RedHat 6.2 CD in drive d: Insert a floppy into drive a: > d: > cd images > ..\dosutils\rawrite -f boot.img -d a -n
Install RedHat 6.2	
Install RedHat Linux version 6.2. Configuration assumes 9GB on device /dev/hda. If larger space is available adjust /home and /var partitions appropriately. Make sure partitions are allocated according to the intented purpose of the server. For example, a syslog server would have more storage dedicated to the /var partition, while a web server would need equal room in both the /var and /home partitions.	
Steps: 1. Boot from floppy or CD. 2. Hit <enter> to select default install at boot: prompt. 3. Select English language. 4. Select default keyboard. 5. Select default mouse. 6. Select Next.	Partitions: / 1000MB root partition /tmp 500MB temporary files /home 3500MB web and ftp files

Select custom installation.
8. Use Disk Druid to partition hard drive(s).
9. Select Check for bad blocks and format partitions.
10. Leave default lilo configuration.
11. Uncheck DHCP option and enter network configuration for each network card in system.
12. Set time zone to UTC and check 'System clock uses UTC'.
13. Enter root password and create admin accounts.
14. Leave default authentication configuration.
15. Clear all bundled packages and choose Select individual packages.
16. Select packages to install.
17. Select Next to begin installation.
18. Create boot disk.
19. Click Exit to finish and reboot.

```
files and mail queue
<swap>  256MB    swap
```

Packages:

Applications

Archiving - dump, zip, unzip
Communications - lrzsz
Editors - jed, jed-common, vim-enhanced
Internet - rsync, traceroute
System - screen, dialog

Development

Debuggers - lsof
Languages - perl, python

System Environment

Base - ipchains, shapecfg
Daemons - anonftp, apache, inetd, iputils, mod_perl, php, tcp_wrappers, wu-ftpd, xntp3
Kernel - kernel-smp
Libraries - freetype
Shell - bash2, tcsh

Shut off runlevel services

Use the chkconfig utility to inspect and turn off services that should not be started at boot time. We will also stop the inet daemon now to minimize the chance of the default services being exploited until we configure the tcp wrappers, ssh and the inet daemon in the coming steps.

Steps:

1. List services that are started at boot time.
2. For each service that you do not want running at startup, use chkconfig to turn it off and then explicitly stop the service.
3. List startup services again to verify.
4. Stop inet daemon temporarily.

```
# chkconfig --list

anacron     0:off    1:off    2:off    3:off
4:off    5:off    6:off
httpd     0:off    1:off    2:off     3:on
4:on    5:on     6:off
apmd     0:off    1:off    2:off     3:on
4:on    5:on     6:off
atd     0:off    1:off    2:off     3:on
4:on    5:on     6:off
keytable     0:off    1:off    2:on     3:on
4:on    5:on     6:off
gpm     0:off    1:off    2:on     3:on
4:on    5:on     6:off
inet     0:off    1:off    2:off     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:on    2:on     3:on
4:on    5:on     6:off
ipchains     0:off    1:off    2:off     3:off
4:off    5:off     6:off
```

```
pcmcia     0:off    1:off    2:off     3:off
4:off    5:off     6:off
kudzu     0:off    1:off    2:off     3:on
4:on    5:on     6:off
linuxconf     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
syslog     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
xntpd     0:off     1:off    2:off    3:on
4:on    5:on    6:off

# chkconfig --del apmd
# /etc/rc.d/init.d/apmd stop
# chkconfig --del netfs
# /etc/rc.d/init.d/netfs stop
# chkconfig --del atd
# /etc/rc.d/init.d/atd stop
# chkconfig --del pcmcia
# chkconfig --del sendmail
# /etc/rc.d/init.d/sendmail stop
# chkconfig --del gpm
# /etc/rc.d/init.d/gpm stop
# chkconfig --del kudzu
# chkconfig --del linuxconf
# chkconfig --del xntpd

# chkconfig --list

anacron     0:off    1:off     2:off    3:off
4:off     5:off    6:off
httpd     0:off    1:off    2:off     3:on
4:on    5:on     6:off
apmd     0:off    1:off    2:off     3:off
4:off    5:off     6:off
atd     0:off    1:off    2:off     3:off
4:off    5:off     6:off
keytable     0:off    1:off    2:on     3:on
4:on    5:on     6:off
gpm     0:off    1:off    2:off     3:off
4:off    5:off     6:off
inet     0:off    1:off    2:off     3:on
4:on    5:on     6:off
netfs     0:off    1:off    2:off     3:off
4:off    5:off     6:off
network     0:off    1:off    2:on     3:on
4:on    5:on     6:off
random     0:off    1:on    2:on     3:on
4:on    5:on     6:off
ipchains     0:off    1:off    2:off     3:off
4:off    5:off     6:off
pcmcia     0:off    1:off    2:off     3:off
4:off    5:off     6:off
kudzu     0:off    1:off    2:off     3:off
4:off    5:off     6:off
linuxconf     0:off    1:off    2:off     3:off
```

	4:off 5:off 6:off sendmail 0:off 1:off 2:off 3:off 4:off 5:off 6:off syslog 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 xntpd 0:off 1:off 2:off 3:off 4:off 5:off 6:off # /etc/rc.d/init.d/inet stop
Configure tcp wrappers	
tcp wrappers are used to limit access to services controlled by inetd. Specifically, we want to deny access to everything except ssh and ftp. SSH will be limited by FQDN (fully qualified domain name) and ftp access will be unlimited. This is done by modifying the files hosts.deny and hosts.allow in the /etc directory. We will also add a line to the hosts.deny configuration file to notify the administrators when failed login attempts occur.	
Steps: 1. Configure hosts.deny. 2. Configure hosts.allow. 3. Run tcpdchk to verify wrapper configuration.	Using an editor, modify the file /etc/hosts.deny. Add a single line to deny everything and send email notification of failed attempts. For example: ALL: ALL: echo "%s: connection attempt from %c" | /usr/sbin/sendmail -f `uname -n` admin@domain.com Now modify the file /etc/hosts.allow. Add a line for each service:host combination that should be allowed access. For example: SSH: host.domain.com in.ftpd: ALL # tcpdchk -v
Configure time synchronization	
Since a secure time server resides on our local network, we will simply set up a cron job to run ntpdate to periodically query that time source. We also run it at boot time to set the clock explicitly rather than waiting for the cron job to fire.	
Steps: 1. Setup cron job to sync time to local time servers. 2. Add initial time synchronization to startup script.	Modify /etc/cron.d/kmod and add the line 3 */4 * * * root /usr/sbin/ntpdate time.domain.com Modify /etc/rc.d/rc.local and add the line /usr/sbin/ntpdate *time.domain.com*
Update RedHat package manager and update packages	
Keeping your system components up to date is half the battle in avoiding known vulnerabilities. New exploits are being announced at least weekly, and fixes usually follow shortly thereafter. For this reason, we will take a few steps to keep on top of updates more managable. RPM is a very good tool for managing the packages installed on a linux server. We will use it, in	

conjunction with the AutoRPM tool by Kirk Bauer to notify administrators by email of new packages that are available.	
Steps: 1. Install the latest package manager from RedHat. 2. Install PGP and RedHat public key. 3. Install AutoRPM. 4. Configure AutoRPM. 5. Run AutoRPM interactively.	# rpm -ivh /mnt/cdrom/rpm-4.0.2-6x.i386.rpm # rpm -ivh gnupg-1.0.4.i386.rpm # gpg # cp /mnt/cdrom/redhat.gpg /root/.gnupg AutoRPM requires the perl-libnet package. # rpm -ivh /mnt/cdrom/perl-libnet-1.0605-2.noarch.rpm # rpm -ivh /mnt/cdrom/autorpm-1.9.8.4-2.noarch.rpm Modify the file /etc/autorpm.d/pools/redhat-updates to limit the list to update sites to updates.redhat.com. Modify the file /etc/autorpm.d/redhat-updates.conf to: • Add the line 'PGP_Require (Yes)' to the section labeled 'action (updated)'. • Change the line Install (Interactive) to Install (No) in the section labeled 'action(new)'. Modify the file /etc/autorpm.d/autorpm.conf to set the ReportDest variable to the email addresses of the administrators. Seperate email addresses with a comma. Set_Var("ReportDest", "admin@domain.com"); Modify the file /etc/cron.daily/autorpm.cron and change the delay value to anything other than the default value. # autorpm --interactive Follow interactive display to update packages. After the initial run, the autorpm.cron script will run daily and send email notification of

	package updates.
Install ssh	
SSH and SCP is used as a secure administration tool for remote access. It provides authentication and encryption while allowing shell access and intra-server file transfers. SSH also provides a means of tunneling other non-secure protocols through it's encrypted, authenticated channel. We will be installing the 1.2.27 version of ssh due to compatibility with legacy clients in place. The truly paranoid would opt for version 2.x, and we should consider upgrading as well. After installing the SSH packages and generating the ssh public and private keys, we will configure the services we want running under the inet daemon. Specifically, we will allow ftp and ssh access here. Then, disable remote root logins for added accountability and change the banners that are used with tcp connections to remove any information that might prove useful to a hacker.	
Steps: 1. Install ssh package. 2. Generate SSH public / private keys. 3. Configure SSH to run under inetd. 4. Turn off unwanted inet services. 5. Change banners. 6. Disable remote root login. 7. Restart inet daemon.	# rpm -ivh /mnt/cdrom/ssh-1.2.27.rpm # ssh-keygen -f /root/.ssh/identity -N '' Edit the file **/etc/inetd.conf** and add the following line. Use <tab> wherever whitespace occurs. ssh stream tcp nowait root /usr/sbin/tcpd /usr/local/sbin/sshd -i While in this same file, ensure every line is commented out, except the line that starts with ftp. When finished, every line in this file should be commented out with the exception of the lines that start with either 'ssh' or 'ftp'. # /etc/rc.d/init.d/inetd restart Modify the files /etc/issue and /etc/issue.net and change them to the company's standard security warning: Hilgraeve Inc. WARNING: Unauthorized use is prohibited. Violators will be prosecuted. Modify the file /etc/sshd_config and change the PermitRootLogin line to read PermitRootLogin no # /etc/rc.d/init.d/inetd restart

Install RedHat secureweb package and configure Apache with mod_ssl	
Install and configure the RedHat secureweb package, which is a bundle of Apache and mod_ssl.	
Steps: 1. Install secureweb package. 2. Install server key and certificate. 3. Configure Apache.	# rpm -ivh /mnt/cdrom/secureweb-3.2-12.i386.rpm # cp /mnt/floppy/server.key /etc/httpd/conf/ssl.key # cp /mnt/floppy/server.crt /etc/httpd/conf/ssl.crt # cp /mnt/floppy/gsid.crt /etc/httpd/conf/ssl.crt Modify the file /etc/httpd/conf/httpd.conf and make the following changes: • Change the user and group that Apache runs under by changing the lines "User nobody" and "Group nobody" to "User web" and "Group web". • Add the line "ServerTokens prod" to minimize Apache header information. • Add the line "SSLCACertificateFile /etc/httpd/conf/ssl.crt/gsid.crt" after the existing "SSLCACertificateFile" line. • Change the line "ServerAdmin root@localhost" to "ServerAdmin admin1@domain.com"
Protected areas of the web site and start Apache	
Secure areas of the web site that will contain sensitive data by requiring username and passwords over an ssl connection to the web server.	
Steps: 1. Create the access file. 2. Create the password file. 3. Set permissions on config files. 4. Start Apache.	Create a file called .htaccess in the directory you wish to secure. The file should contain the following information: AuthName "www.domain.com" AuthType Basic AuthUserFile /etc/httpd/conf/httpusers require valid-user #leave out for 'all' access SSLRequireSSL # htpasswd -c /etc/httpd/conf/httpusers *username*

	Enter and confirm the password for the new user. Repeat for all users. # chmod -R o-rwx /etc/httpd/conf/* /etc/httpd/conf # chown nobody.nobody /etc/httpd/conf/htpasswd # /etc/rc.d/init.d/httpsd start
Configure Syslog	
Configure the rotation of the system, ftp and web logs to occur daily. This is for convenience sake in that huge log files are difficult to manage. Configure logrotate to keep a year's worth of compressed logs on the system. Syslog is then configured to redirect critical system messages to a remote syslog server elsewhere on the local network.	
Steps: 1. Configure log rotation. 2. Send important log messages to remote syslog server. 3. Restart syslogd.	Modify the file /etc/logrotate.conf and make the following changes • Change rotation frequency to 'daily'. • Keep 356 days of logs. • Uncomment the 'compress' line. Modify the file /etc/syslog and set remote logging options for messages you want to send to the syslog server for inspection. For example: # Log anything (except mail) of level info or higher. # Don't log private authentication messages! *.info;mail.none;authpriv.none @192.168.1.5 # The authpriv file has restricted access. authpriv.* @192.168.1.5 # Log all the mail messages in one place. mail.* @192.168.1.5 # /etc/rc.d/init.d/syslog restart
Install and run Tiger (TAMU)	
Tiger is a set of bash shell scripts which will perform a security audit of the system. The result is a report of possible ways the root account could be compromised. While everything listed would not necessarily need to be fixed, the list should be reviewed to determine which vulnerabilities should be fixed.	
Steps: 1. Untar tiger.	# cd /usr/local/src # tar -xzvf /mnt/cdrom/tiger-2.2.4p1.tar.gz

Run tiger with default tigerrc. 3. Review resulting report and determine which items need to be fixed. 4. Fix items identified in step 3.	# ./tiger
Install and configure Tripwire	
Tripwire is a security tool used for intrusion detection and filesystem integrity checking. It will store an encrypted database of the files on the system and when run via a cron job will notify administrators if anything changes on the system.	
Steps: 1. Install Tripwire. 2. Configure Tripwire. 3. Run Tripwire to baseline system. 4. Setup Tripwire as a cron job.	# rpm -ivh /mnt/cdrom/tripwire-2.3-47.i386.rpm # cp /mnt/cdrom/twpol.txt /etc/tripwire # /etc/tripwire/twinstall.sh Record passphrases: Site keyfile phrase:_____ Local keyfile phrase:_____ # tripwire --init # tripwire --check Modify the file /etc/cron.daily/kmod and add the line: 3 */4 * * * root /usr/sbin/tripwire --check > /dev/null
Scan system for vulnerabilities	
Scan the server from inside and out. ps lists running processes on the system. Use it to inspect the list of processes now to verify that we have shut down all unnecessary services. lsof is a very useful security tool. It can be used to investigate any processes currently running on the system. Here we will verify only the ports we expect are listening for connections. Nmap is a port scanning tool which is used from another computer residing on the local network with the server we're building. It gives us a hacker's view of our system in that it will list open ports on the system. It also attempts to guess the operating system, which in our case does a good job.	
Steps: 1. Use ps to verify system processes. 2. Run lsof to verify no unexpected ports are open. 3. Use nmap to scan system for open ports.	# ps -A The ps output should look like this: PID TTY TIME CMD 1 ? 00:00:08 init 2 ? 00:00:00 kflushd 3 ? 00:00:01 kupdate 4 ? 00:00:00 kpiod 5 ? 00:00:00 kswapd 6 ? 00:00:00 mdrecoveryd 291 ? 00:00:03 syslogd 300 ? 00:00:03 klogd 314 ? 00:00:00 crond

```
328 ? 00:00:00 inetd
434 tty1 00:00:00 mingetty
435 tty2 00:00:00 mingetty
436 tty3 00:00:00 mingetty
437 tty4 00:00:00 mingetty
438 tty5 00:00:00 mingetty
439 tty6 00:00:00 mingetty
534 pts/0 00:00:00 bash
607 ? 00:00:00 in.ftpd
608 ? 00:00:15 sshd
712 ? 00:00:08 httpsd
713 ? 00:00:03 httpsd
714 ? 00:00:03 httpsd
715 ? 00:00:03 httpsd
718 ? 00:00:03 httpsd
1000 pts/0 00:00:00 ps
```

lsof -i

The lsof output should look like this:

```
COMMAND PIDg USER FD TYPE
DEVICE SIZE NODE NAME
inetd 492 root 4u IPv4 476 TCP *:ssh
(LISTEN)
inetd 11309 root 4u IPv4 15525 TCP *:ftp
(LISTEN)
httpsd 23545 root 16u IPv4 6618745 TCP
*:https (LISTEN)
httpsd 23545 root 17u IPv4 6618746 TCP
*:www (LISTEN)
```

nmap -sS -O 192.168.1.54

The nmap output should look like this:

```
Starting nmap V. 2.53 by
fyodor@insecure.org (
www.insecure.org/nmap/ )
  Interesting ports on (192.168.1.54):
(The 1518 ports scanned but not shown
below are in state: closed)
Port State Service
21/tcp open ftp
22/tcp open ssh
80/tcp open http
443/tcp open https

TCP Sequence Prediction: Class=random
positive increments
Difficulty=3239902 (Good luck!)
Remote operating system guess: Linux
2.1.122 - 2.2.14
```

Create spare drive for system

Install an identical second drive in the system. Go into single-user mode and use the dd command to copy disk to disk. Then, go back into multi-user mode and verify copy. Finally, run fsck on all partitions to verify integrity. Assuming your disk configuration is IDE, the commands would look like below.

Steps: 1. Install second drive in system. 2. Clone main drive for backup purposes. 3. Remove second hard drive.	# init 1 # dd if=dev/hda of=/dev/hdb bs=lk # init 3 # fdisk -l /dev/hdb # fsck /dev/hdb1 # fsck /dev/hdb[n]...

Data Center Installation and Ongoing Support	
Install server in data center	
Assuming we've loaded the web and ftp content, it's time to reconfigure the network settings and move the server to the data center.	
Steps: 1. Reconfigure network settings 2. Pack server and deliver server to data center. 3. Install server the cabinet. 4. Verify connectivity and test web site. 5. Lock cabinet. 6. Distribute data center access cards and cabinet keys to primary and secondary administrators of the system.	Modify the file /etc/resolv.conf and enter the dns settings for the data center network. nameserver ns1.domain.com nameserver ns2.domain.com Modify the file /etc/sysconfig/network and change the gateway setting. GATEWAY=192.168.1.1 Modify the files /etc/sysconfig/network-scripts/ifcfg-eth[n] and change the IP address settings. IPADDR=192.168.1.54 NETMASK=255.255.255.128
Secure documents	
Needless to say, this document, and the contents of the archive drive we made are invaluable to a hacker wishing to compromise this system. Not only do they contain sensitive information, they represent the tools we have available to us to recover this system in the case of an emergency. They should be treated as such and secured in a location seperate from the servers.	
Steps: 1. Secure items offsite.	Place the following items in a secure location offsite: • This document. • Server key and certificate diskette • Spare cloned drive • System boot diskette
Ongoing Support	
New vulnerabilities in systems components are being discovered and exploited daily. Keeping up-to-date on the latest vulnerabilities and taking appropriate action to thwart them is the only way to guarantee the security of any system. To help stay abreast of current issued related to systems security, Systems Administrators should subscribe to several list servers dealing with applicable security related topics. Listed below is a basic list of URLs that should cover most everything security related with regards to the RedHat Linux operating system:	
Steps:	Security-related mailing lists:

Subscribe to security mailing lists. 2. React quickly to security alerts that relate to the system.	http://www.securityfocus.com **RedHat** Security Advisories https://listman.redhat.com/mailman/listinfo/redhat-watch-list **RedHat** bugfix announcements https://listman.redhat.com/mailman/listinfo/redhat-watch-list **CERT** Advisories http://www.cert.org/advisories

References

[1] Gray, Michael "Build a Secure Web Server Using Red Hat Linux Version 6.2", http://www.sans.org/y2k/practical/Michael_Gray_GCUX.doc

[2] Brotzman, Lee E. and Ranch, David A. Securing Linux Step-By-Step Version 1.0, The SANS Institute, 2000.

[3] Garfunkel, Simson and Spafford, Gene Practical UNIX & Internet Security Second Edition, O'Reilly & Associates, 1996