Global Information Assurance Certification Paper

Secure Linux RedHat 7 FTP Anonymous Upload Server Step by Step

Rui Wang

Executive Summary

The Biostats computer system serves as the data coordinating center for numerous NIH grant studies. There are two programs running on Biostats system automatically process the data and insert it into the Oracle Database. The data is uploaded from two outside research centers using Biostats FTP server. These two outside FTP connections make Biostats system vulnerable to a password scanning attack, FTP service attack, denial of service attack...

To resolve this security problem, I propose to setup a secure RedHat 7 anonymous FTP upload server to accept data from these two outside research centers. This FTP server will not use Biostats authentication service, and will not be trusted by any Biostats system. After outside research centers upload data, Biostats computer system picks up data from the FTP server through SSH channel, and then process it.

This report shows the details of the system setup.

The major steps I have taken are: install Operating System, run up2date, run Bastille, verify the general system security, configure FTP server, configure SSH, run Tripwire, run backup, bring system online.

1. Install Operating System

Hardware configuration: Gateway Celeron 333MHz processor, 96M RAM, 3G hard drive, CD-ROM, floppy.

During installation and configuration process, this machine is completely behind firewall. The machine is located in the Biostats secured computer server room with all of Biostats servers, so I do not worry about physical security.

I install Linux from the RedHat 7 CD.

Language selection: English
Keyboard configuration: Generic 101 key PC
Mouse configuration: Generic 3 Button Mouse (PS/2), Emulate 3 Buttons
Install type: Custom System
Manually partition with Disk Druid:

```
/       1000MB
/var    1000MB
/home   800MB
<swap>  200MB
```

The /var partition contains log files and FTP root directory.

The /home contaits users' home directories. The system's cronjob will move uploaded data to two different users' home directories, so the Biostats system can pick them up. This partition also serves as intermediate media for the bare-metal backup file.

Lilo configuration: create boot disk

Install LILO boot record on: /dev/hda Master Boot Record(MBR)
Default boot image:
partition:/dev/hda1 type: Linux nativeboot lable: linux

Network configuration:

```
eth0
activate on boot
ip address: 169.146.60.136
netmask:        255.255.255.0
network:        169.146.60.0
Broadcast:      169.146.60.255

Hostname:       ward
Gateway:        169.146.60.253
Primary DNS:    169.146.60.186
Secondary DNS:  169.146.60.126
```

Time Zone: Eastern time

Account Configuration: root, rwang(that's my account), hemo and vaslab

Authentication configuration:

enable MD5 passwords
enable shadow passwords

Package Group selection:

printer support
X Window system
Gnome
Mail/WWW/News Tools
Networked Workstation
Utilities

I choose to select individual packages:
Under applications -> archiving, I add pax, unzip,zip.
Under applications -> system, I add tripwire.
Under development -> debugger, I add lsof.
Under applications -> Internet, I keep these packages only: ftp, Netscape, openssh, pine, whois, traceroute, tcpdump, stunnel.
Under system environment -> Base, I keep up2date, up2date-gnome.
Under system environment -> Daemons, I keep wu-ftpd xinetd tcp_wappers, ntp

I am sure I still have more packages than I need, but I will make sure they won't hurt system security by running up2date, Bastille and tripwire.

I choose to skip X configuration for now.

2. Run up2date

RedHat puts information about bug fixes on its "errata" page. It is important to apply all updated bug fixes to the system. RedHat Update Agent - up2date is a program to check, download and install updated RPMS.

Up2date has X Window GUI interface, and can be run from command line too. I would like to use GUI when I run it the first time. I choose to run X Window GUI programs from SSH remote login window.

> Notice that I don't want to run X Window on the console, but I do want to configure it right, just in case I need it in the future.
>
> To configure X, login on the console as root, run Xconfigurator to make X Window work, but do not start up X

at boot. To start up X Window, type in command "init 5", runlevel 5 starts X Window server and uses X Display Manager login. To go back to character-mode, type in "init 3", runlevel 3 is the default runlevel on this machine.

I also want to configure a printer using printtool. It is convenient to have a printer ready for me. And I may need to use it when I do troubleshooting, or incident investigation later, even though I will not start up lpd at boot.

Run /usr/sbin/up2date-config to configure up2date.
Don't display packages when local configuration file has been modified
Use GPG to verify package integrity
Package names to skip: kernel*

Run /usr/sbin/up2date. Install RedHat public key. Then let the program retrieve update packages and install them.

I need to run this program once a week. To configure it to run in "batch mode" as a cronjob every Saturday at 2:02AM,

```
# cat > /etc/cron.d/up2date_cron
2 2 * * 6       root    /usr/sbin/up2date -u
^D
```

3. Run Bastille

Bastille is a program which can be used to "harden" or "tighten" the Linux operating system. I download it from http://www.bastille-linux.org/.

I Run the text user interface to generate a configuration file, and implement the changes after.

```
# /root/Bastille/InteractiveBastille.pl
```

When I run it for the first time on RedHat 7, it complains about libncurses.so.4 file not found. I make a link under /use/lib, libncurses.so.4 -> libncurses.so.5.1, and it runs.

With this program, I take off many system programs' SUID bit, run sendmail as cronjob instead of daemon, close out all unneeded xinetd services...

Here is the set of interesting questions and my answers.
Q: Would you like to run the ipchains script? No (we have firewall already)
Q: Would you like to download and install the updated RPMs? No (I run up2date)
Q: Would you like to set more restrictive permissions on the administration utilities? Yes
Q: Would you like to disable SUID status for mount/umount, ping, dump, restore, cardctl, at, DOSMU? Yes
Q: Would you like to disable SUID status for news server tools? Yes
Q: Would you like to disable SUID status for r-tools? Yes
Q: Would you like to disable SUID status for usernetctl? Yes
Q: Would you like to disable SUID status for printing utilities? Yes
Q: Would you like to disable SUID status for traceroute? Yes
Q: Would you like to set up a second UID 0 account? No
Q: May we take strong steps to disallow the dangerous r-protocols? Yes
Q: Would you like to enforce password aging? Yes
Q: Would you like to create a non-root user account? No (I use linuxConf to create user accounts)
Q: Would you like to restrict the use of cron to administrative accounts? Yes
Q: Would you like to password-protect the LILO prompt? Yes
Q: Would you like to reduce the LILO delay time to zero? Yes
Q: Do you ever boot Linux from your hard drive? Yes
Q: Would you like to write the LILO changes to a boot floppy? No
Q: Would you like to disable CTRL-ALT-DELETE rebooting? Yes
Q: Would you like to password protect single-user mode? Yes
Q: Would you like to modify inetd.conf and /etc/hosts.allow to optimize use of Wrappers? Yes

Q: Would you like to set sshd to accept connections only from a small list of IP addresses? Yes
Q: Would you like to make "Authorized Use" banner? Yes
Q: Would you like to disable the compiler? Yes
Q: Would you like to put limits on system resource usage? Yes
Q: Should we restrict console access to a small group of user accounts? Yes (I input root and rwang)
Q: Would you like to add additional logging? Yes
Q: Do you have a remote logging host? Yes (I input Biostats' logging server IP address)
Q: Would you like to set up process accounting? No
Q: Would you like to disable apmd, atd, PCMCIA services, DHCP daemon, GPM, news server daemon, SNMPD? Yes
Q: Would you like to deactivate NFS and Samba, the routing daemons, NIS server and client programs? Yes
Q: Do you want to leave sendmail running in daemon mode? No
Q: Would you like to run sendmail via cron to process the queue? Yes
Q: Would you like to disable the VRFY and EXPN sendmail commands? Yes
Q: Would you like to download and install ssh? No (I have SSH already)
Q: Would you like to deactivate named and the Apache web server? Yes

4. Verify the general system security changes with command more, ps, nmap

a. /etc/lilo.conf file

```
# more /etc/lilo.conf
restricted
password=aaa(my lilo password)
delay=1
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
#prompt
#timeout=50
message=/boot/message
linear
default=linux

image=/boot/vmlinuz-2.2.16-22
        label=linux
        read-only
        root=/dev/hda1
```

Bastille changes this file to reduce the LILO delay time to zero and password-protect the LILO prompt. It looks good to me. Set /etc/lilo.conf file immutable:

```
# chattr +i /etc/lilo.conf
```

b. /etc/inittab file

```
# more /etc/inittab

id:3:initdefault:

~~:S:wait:/sbin/sulogin

si::sysinit:/etc/rc.d/rc.sysinit

l0:0:wait:/etc/rc.d/rc 0
l1:1:wait:/etc/rc.d/rc 1
l2:2:wait:/etc/rc.d/rc 2
l3:3:wait:/etc/rc.d/rc 3
l4:4:wait:/etc/rc.d/rc 4
```

```
l5:5:wait:/etc/rc.d/rc 5
l6:6:wait:/etc/rc.d/rc 6

ud::once:/sbin/update

#ca::ctrlaltdel:/sbin/shutdown -t3 -r now

pf::powerfail:/sbin/shutdown -f -h +2 "Power Failure; System Shutting Down"
pr:12345:powerokwait:/sbin/shutdown -c "Power Restored; Shutdown Canceled"

1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6

x:5:respawn:/etc/X11/prefdm -nodaemon
```

Bastille changes this file to password protect single-user mode and disable CTRL-ALT-DELETE rebooting. Set /etc/inittab file immutable:

```
# chattr +i /etc/inittab
```

c. /etc/hosts.allow and /etc/hosts.deny files:

RedHat 7 uses tcp-wrapper and xinetd by default for all inetd services. I do need to make some changes here:

```
# more /etc/hosts.deny
ALL:ALL@ALL

# more /etc/hosts.allow
sshd: 169.146.60. 127.0.0.1 : ALLOW
in.ftpd: 169.146.60. 12.161.125.18 152.73.145.21 : ALLOW
```

Note: 169.146.60. is Biostats internal network, 12.161.125.18 is HEMO research center's IP address and 152.73.145.21 is Vaslab's IP address.

This says allow ssh connection from Biostats internal network only, ftp connection from Biostats internal network and two research centers only, deny everything else.

The format of xinetd's configuration files is different from /etc/inetd.conf file:

```
# more /etc/xinetd.conf
defaults
{
        instances               = 60
        log_type                = SYSLOG authpriv
        log_on_success          = HOST PID
        log_on_failure          = HOST RECORD
}
includedir /etc/xinetd.d
```

Under /etc/xinetd.d directory, each inetd service has its own configuration file. If you want to manually disable one, add a line "disable = yes" in its configuration file. Here is an example:

```
# more /etc/xinetd.d/echo
service echo
{
        type            = INTERNAL
        id              = echo-stream
        socket_type     = stream
        protocol        = tcp
        user            = root
```

```
        wait            = no
        disable         = yes
}
```

The only service should be enabled is wu-ftpd:

```
# more /etc/xinetd.d/wu-ftpd
service ftp
{
        socket_type             = stream
        wait                    = no
        user                    = root
        server                  = /usr/sbin/in.ftpd
        server_args             = -l -a
        log_on_success          += DURATION USERID
        log_on_failure          += USERID
        nice                    = 10
}
```

d. /etc/security/access.conf, /etc/security/limits.conf and /etc/security/time.conf files:

```
# more /etc/security/access.conf
-:ALL EXCEPT root rwang:console
```

This says only root and rwang(that's me) are allowed to login from the console.

```
# more /etc/security/limits.conf
*       hard    core    0
*       soft    nproc   100
*       hard    nproc   150
*       hard    fsize   40000
```

This says to prohibit the creation of core files, restrict the number of processes to a soft limit of 100 and a hard limit of 150 for everyone except the super-user root, limit size of any one of users' files to 40mb.

/etc/security/time.conf file limits when users can login. I don't want put on time restriction, so I leave it empty.

e. /etc/host.conf file:

This file specifies how names are resolved. I change this file to be:

```
order bind,hosts
multi on
nospoof on
```

It says lookup names via DNS first then /etc/hosts file, allow machine with multiple IP addresses, not allow IP spoofing.

f. SUID and SGID files:

Bastille takes care of some SUID files, let me find all SUID and SGID files again:

```
# find / -type f \( -perm -04000 -o -perm -02000 \) -exec ls -lg {} \;
-rwsr-xr-x    1 root     root        14184 Jul 12 20:47 /bin/su
-rwxr-s---    1 root     root         4116 Aug 23 22:41 /sbin/netreport
-r-sr-xr-x    1 root     root        14732 Aug 22 21:19 /sbin/pwdb_chkpwd
-r-sr-xr-x    1 root     root        15340 Aug 22 21:19 /sbin/unix_chkpwd
-rws--x--x    1 root     root         6024 Aug 30 15:55 /usr/X11R6/bin/Xwrapper
-rwsr-xr-x    1 root     root        21248 Aug 24 18:30 /usr/bin/crontab
-rwsr-xr-x    1 root     root        34220 Aug  8 11:40 /usr/bin/chage
-rwsr-x---    1 root     root        36344 Aug  8 11:40 /usr/bin/gpasswd
-r-xr-sr-x    1 root     tty          6524 Aug  8 19:52 /usr/bin/wall
-rwsr-xr-x    1 lp       lp         444512 Sep 25 10:53 /usr/bin/lpr
```

```
-rwxr-sr-x    1 root     man         35260 Aug 23 11:56 /usr/bin/man
-rwsr-xr-x    1 root     root       155804 Oct  5 18:09 /usr/bin/ssh
-r-s--x--x    1 root     root        13536 Jul 12 07:56 /usr/bin/passwd
-rws--x--x    2 root     root       793603 Aug  7 11:05 /usr/bin/suidperl
-rws--x--x    2 root     root       793603 Aug  7 11:05 /usr/bin/sperl5.6.0
-rwxr-sr-x    1 root     mail        10932 Aug 11 11:36 /usr/bin/lockfile
-rwsr-sr-x    1 root     mail        63772 Aug 11 11:36 /usr/bin/procmail
-rwxr-sr-x    1 root     slocate     23964 Aug 23 21:03 /usr/bin/slocate
-rws--x--x    1 root     root        13184 Aug 30 17:54 /usr/bin/chfn
-rws--x--x    1 root     root        12640 Aug 30 17:54 /usr/bin/chsh
-rws--x--x    1 root     root         5464 Aug 30 17:54 /usr/bin/newgrp
-rwxr-sr-x    1 root     tty          8500 Aug 30 17:54 /usr/bin/write
-rwsr-x---    1 root     root         6288 Aug 23 22:41 /usr/sbin/usernetctl
-rwxr-sr-x    1 root     utmp         9212 Aug 23 20:46
/usr/sbin/gnome-pty-helper
-rwxr-s---    1 lp       lp         435008 Sep 25 10:53 /usr/sbin/lpc
-r-sr-xr-x    1 root     root       401748 Aug 22 21:32 /usr/sbin/sendmail
-rwsr-x---    1 root     root        20880 Oct  6 10:32 /usr/sbin/userhelper
-rwxr-sr-x    1 root     utmp         6584 Jul 13 00:46 /usr/sbin/utempter
```

That's still too many. Let me take care of some manually:

```
# chmod a-s /usr/bin/lockfile /usr/X11R6/bin/Xwrapper /usr/sbin/usernetctl
# chmod a-s /usr/sbin/gnome-pty-helper /usr/sbin/lpc /usr/sbin/userhelper
# chmod a-s /usr/bin/chfn /usr/bin/chsh /usr/bin/newgrp
# chmod a-s /usr/bin/write /usr/bin/procmail /sbin/netreport /usr/bin/chage
# chmod a-s /usr/bin/gpasswd /usr/bin/wall /usr/bin/lpr
```

Now, check all SUID and SGID files again:

```
#  find / -type f \( -perm -04000 -o -perm -02000 \) -exec ls -lg {} \;
-rwsr-xr-x    1 root     root        14184 Jul 12 20:47 /bin/su
-r-sr-xr-x    1 root     root        14732 Aug 22 21:19 /sbin/pwdb_chkpwd
-r-sr-xr-x    1 root     root        15340 Aug 22 21:19 /sbin/unix_chkpwd
-rwsr-xr-x    1 root     root        21248 Aug 24 18:30 /usr/bin/crontab
-rwxr-sr-x    1 root     man         35260 Aug 23 11:56 /usr/bin/man
-rwsr-xr-x    1 root     root       155804 Oct  5 18:09 /usr/bin/ssh
-r-s--x--x    1 root     root        13536 Jul 12 07:56 /usr/bin/passwd
-rws--x--x    2 root     root       793603 Aug  7 11:05 /usr/bin/suidperl
-rws--x--x    2 root     root       793603 Aug  7 11:05 /usr/bin/sperl5.6.0
-rwxr-sr-x    1 root     slocate     23964 Aug 23 21:03 /usr/bin/slocate
-r-sr-xr-x    1 root     root       401748 Aug 22 21:32 /usr/sbin/sendmail
-rwxr-sr-x    1 root     utmp         6584 Jul 13 00:46 /usr/sbin/utempter
```

That's better.

g. sendmail daemon:

Bastille asked questions about sendmail daemon, but it didn't take care of this. So I do this manually.

```
# /etc/rc.d/init.d/sendmail stop
# chkconfig --del sendmail
# cat > /etc/cron.d/sendmail_cron
10 * * * *      root    /usr/sbin/sendmail -q
^D
```

In the meantime, I take care of other unused daemon:

```
# /etc/rc.d/init.d/kudzu stop
# chkconfig --del kudzu
# /etc/rc.d/init.d/lpd stop
# chkconfig --del lpd
# /etc/rc.d/init.d/xfs stop
# chkconfig --del xfs
# /etc/rc.d/init.d/identd stop
# chkconfig --del identd
```

```
# /etc/rc.d/init.d/rhnsd stop
# chkconfig --del rhnsd
```

Forward root email to Biostats system group by adding this line in /etc/aliases file:

```
root: system@biostats.ccf.org
```

Send all ftp, ssh connections, boot messages to loghost, Biostats logging server. Edit /etc/syslog.conf file and add this line:

```
authpriv.*;local7.*;auth.*;daemon.info  @loghost.biostats.ccf.org
```

Synchronize time with Biostats NTP server by running ntpdate at cron:

```
# cat > /etc/cron.hourly/ntp_cron
#!/bin/sh

/usr/sbin/ntpdate -su -t 3 timeserver.biostats.ccf.org
/sbin/hwclock --systohc
^D
# chmod 700 /etc/cron.hourly/ntp_cron
```

Set login time out for the root account:

```
# echo TMOUT=7200 >> /etc/profile
```

h. the networking security kernel tunable parameters

RedHat 7 uses /etc/sysctl.conf file to set kernel parameters. I edit /etc/sysctl.conf file, add some networking security options there. The comment lines explain the parameters pretty well.

```
# more /etc/sysctl.conf
# Disables packet forwarding
net.ipv4.ip_forward = 0
# Enables source route verification
net.ipv4.conf.all.rp_filter = 1
# Disables automatic defragmentation (needed for masquerading, LVS)
net.ipv4.ip_always_defrag = 0
# Disables the magic-sysrq key
kernel.sysrq = 0
# Enable ignoring ping request
net.ipv4.icmp_echo_ignore_all = 1
# Enable ignoring broadcasts request
net.ipv4.icmp_echo_ignore_broadcasts = 1
# Disables IP source routing
net.ipv4.conf.all.accept_source_route = 0
# Enable TCP SYN Cookie Protection
net.ipv4.tcp_syncookies = 1
# Disable ICMP redirect acceptance
net.ipv4.conf.all.accept_redirects = 0
# Enable bad error message protection
net.ipv4.icmp_ignore_bogus_error_responses = 1
# Log spoofed packets, source routed packets, redirect packets
net.ipv4.conf.all.log_martians = 1
```

i. reboot the machine, check the output of ps command and output of nmap command from its neighbor

```
$ ps -A
  PID TTY          TIME CMD
    1 ?        00:00:05 init
    2 ?        00:00:00 kflushd
    3 ?        00:00:00 kupdate
    4 ?        00:00:00 kpiod
    5 ?        00:00:00 kswapd
    6 ?        00:00:00 mdrecoveryd
```

```
   68 ?        00:00:00 khubd
  328 ?        00:00:00 syslogd
  338 ?        00:00:00 klogd
  390 ?        00:00:00 xinetd
  426 ?        00:00:00 sshd2
  440 ?        00:00:00 crond
  473 tty1     00:00:00 mingetty
  474 tty2     00:00:00 mingetty
  475 tty3     00:00:00 mingetty
  476 tty4     00:00:00 mingetty
  477 tty5     00:00:00 mingetty
  478 tty6     00:00:00 mingetty
  481 ?        00:00:00 sshd2
  485 pts/0    00:00:00 bash
  503 pts/0    00:00:00 ps
```

Output of nmap from its neighbor:

```
# nmap -sS -O ward

Starting nmap V. 2.3BETA14 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Interesting ports on ward (169.146.60.118):
Port    State       Protocol  Service
21      open        tcp       ftp
22      open        tcp       ssh

TCP Sequence Prediction: Class=random positive increments
                         Difficulty=5913522 (Good luck!)
Remote operating system guess: Linux 2.1.122 - 2.2.13

Nmap run completed -- 1 IP address (1 host up) scanned in 3 seconds
```

This is clean enough for me.

5. Configure anonymous FTP server:

This anonymous FTP server is setup to upload files only. Once the file is uploaded, it should be well protected, so no one except root and user hemo or user vaslab should be able to read the file or the file name. HEMO research center's upload files are owned by user hemo, Vaslab's files are owned by user vaslab. They will be moved to /home/hemo or /home/vaslab, and ready for user hemo or vaslab login from Biostats system to pick up. This FTP server only accepts connections from those two outside research centers and the internal network, and only accepts file name that does not start with "." or "-", and has a length of one to twenty characters. Total hemo's files cannot exceed 20000 blocks, 1000 inodes. Total vaslab's files cannot exceed 20000 blocks, 1000 inodes. It should not show the FTP daemon's version information.

/etc/ftpusers file contains a list of all the users that are not allowed to log in via FTP. Put every username of /etc/passwd file there first, then remove ftp, rwang(my user account).

```
# more /etc/ftpusers
root
bin
daemon
adm
lp
sync
shutdown
halt
mail
news
uucp
operator
games
nobody
```

```
hemo
vaslab
```

The /hosts.allow and /hosts.deny files limit ftp connections from those two outside research centers and internal network only. They are in section 4.

The FTP home directory is /var/ftp. The anonftp package that comes with RedHat 7 sets up the anonymous FTP home directory with the correct permissions, password file, etc.

```
# ls -la /var/ftp
total 24
drwxr-xr-x    6 root     root         4096 Oct 25 12:12 .
drwxr-xr-x   18 root     root         4096 Dec  7 13:47 ..
d--x--x--x    2 root     root         4096 Oct 25 12:12 bin
d--x--x--x    2 root     root         4096 Oct 25 12:12 etc
drwxr-xr-x    2 root     root         4096 Oct 25 12:12 lib
drwxr-sr-x    2 root     ftp          4096 Aug 17 10:53 pub
```

Create two incoming directories without read access:

```
# mkdir -m 733 /var/ftp/hemo
# chown hemo /var/ftp/hemo
# mkdir -m 733 /var/ftp/vaslab
# chown vaslab /var/ftp/vaslab
```

Use LinuxConf (/sbin/linuxconf) to enable user quota on /var. Use command edquota to set quota for user hemo and vaslab on /var. This will restrict total size and total number of upload files.

Edit /etc/ftpaccess file for these reasons:

a. Protect a writable directory from changes:

```
chmod           no              guest,anonymous
delete          no              guest,anonymous
overwrite       no              guest,anonymous
rename          no              guest,anonymous
umask           no              guest,anonymous
```

b. Change greeting for FTP service, this will hide FTP daemon version information. Add this line:

```
greeting text FTP server ready.
```

c. Restrict upload file names, they can have hyphen, upper and lowercase letters, digits, period and underscore, with length from 1 to 20 characters. File names can not start with a period or hyphen:

```
path-filter     anonymous /etc/pathmsg ^[-A-Za-z0-9\._]{1,20}$ ^\. ^-
```

d. Set final disposition of uploaded files, files UID hemo for files under /var/ftp/hemo/, UID vaslab for files under /var/ftp/vaslab/, GID ftp, mode 0600, and no subdirectories; files are "write-only", no downloads:

```
upload /var/ftp /hemo   yes hemo ftp 0600 nodirs
noretrieve /var/ftp/hemo/

upload /var/ftp /vaslab   yes vaslab ftp 0600 nodirs
noretrieve /var/ftp/vaslab/
```

When HEMO research center uploads a file to /var/ftp/hemo, the file is owned by user hemo. I will have a cronjob running as user hemo to move this file to /home/hemo. User hemo logins from Biostats system through SSH channel to pick it up. Vaslab's files will be taken care of in the same way.

e. Deny upload and download privileges for other FTP home areas:

```
upload  /var/ftp/*      /       no
upload  /var/ftp/*      /etc    no
upload  /var/ftp/*      /dev    no
upload  /var/ftp/*      /bin    no
upload  /var/ftp/*      /lib    no
upload  /var/ftp/*      /pub    no

noretrieve      /var/ftp/etc
noretrieve      /var/ftp/dev
noretrieve      /var/ftp/bin
noretrieve      /var/ftp/lib
```

6. Configure SSH

I use ssh-2.3.0 on other Biostats servers, so it is easier for me to use the same package on this machine. The package for Linux version is ssh-commercial-server-2.3.0-1nox.i386.rpm, and was downloaded from http://www.ssh.com. I have SSH Secure Shell Site License.

I remove all openssh packages first,

```
# /etc/rc.d/init.d/sshd stop
# rpm -e openssh-server-2.2.0p1-5 openssh-2.2.0p1-5
# rpm -e openssh-askpass-gnome-2.2.0p1-5 openssh-askpass-2.2.0p1-5
# rpm -e openssh-clients-2.2.0p1-5
```

Add ssh-commercial-server-2.3.0:

```
# rpm -ihv --nodeps ssh-commercial-server-2.3.0-1nox.i386.rpm
```

The system files for SSH2 are by default in /etc/ssh2. Sshd2 is in /usr/local/sbin, All the other binaries are in /usr/local/bin.

The system public key pair: /etc/ssh2/hostkey and /etc/ssh2/hostkey.pub. Generate the hostkey with following command:

```
# ssh-keygen2 -P /etc/ssh2/hostkey
```

The configuration files for client and server, respectively: /etc/ssh2/ssh2_config and /etc/ssh2/sshd2_config.

The system-wide recognized hostkeys are stored under /etc/ssh2/knownhosts.

This SSH server serves for two purposes, one for remote login from Biostats system, one for dvorak(one of Biostats machines) automatically picks up uploaded files. I will use host-based authentication for dvorak, password authentication for other machines, and no root logins.

First, I put this line in both machines /etc/ssh2/ssh2_config file, all ssh requests should tell sshd the clinet's domain name:

```
        DefaultDomain                   biostats.ccf.org
```

I edit /etc/ssh2/sshd2_config file, these two lines make sshd work for Biostats internal machines only:

```
        RequireReverseMapping           yes
        AllowHosts                      *.biostats.ccf.org
```

These three lines make dvorak host-based authentication possible:

```
        AllowedAuthentications          hostbased,password
        AllowSHosts                     dvorak.biostats.ccf.org
        IgnoreRhosts                    no
```

These two lines disables all root logins:

```
IgnoreRootRHosts                yes
PermitRootLogin                 no
```

This line says do not use user's $HOME/.ssh2/knownhosts/ directory to fetch hosts public keys when using host-based authentication.

```
UserKnownHosts                  no
```

Start up sshd and make it start up at boot:

```
# chkconfig --add sshd
# /etc/rc.d/init.d/sshd start
```

Then I copy dvorak's /etc/ssh2/hostkey.pub file to ward(this Linux machine)'s /etc/ssh2/knownhosts/dvorak.biostats.ccf.org.ssh-dss.pub file:

```
# scp rwang@dvorak://etc/ssh2/hostkey.pub /etc/ssh2/knownhosts/dvorak.biostats.ccf.org.ssh-dss.pub
```

I put a .shosts file under my home directory(/home/rwang):

```
dvorak.biostats.ccf.org rwang
```

User hemo's /home/hemo/.shosts file is:

```
dvorak.biostats.ccf.org hemo
```

User vaslab's /home/vaslab/.shosts file is:

```
dvorak.biostats.ccf.org vaslab
```

Make sure these .shosts files are owned by the corresponding users, and have permissions 400.

Now, user hemo, vaslab and rwang don't need to provide passwords when ssh or scp from dvorak to ward(this Linux machine).
Here is an example: When user hemo on dvorak needs to pick up a uploaded file on ward, he uses this command to find the file name:

```
% ssh ward ls
```

Then he uses this command to copy the file over and remove the source file:

```
% scp -u hemo@ward:filename filename
```

7. Run tripwire

Tripwire is a file integrity checker for UNIX systems. I install it from a RedHat 7 CD #2. Tripwire software works by first scanning a computer and creating a database of system files, a compact digital "snapshot" of the system in a known secure state. Once this baseline database is created, I can use the database file as the baseline for integrity checking.

I take the default configuration and policy files for now. I create Tripwire site and local passphrases and use them to sign configuration and policy files:

```
# /etc/tripwire/twinstall.sh
```

Generate the database:

```
# tripwire --init
```

The database file is /var/lib/tripwire/ward.twd. I copy it to a floppy disk to secure it from being modified. It is

important because data from Tripwire is only as trustworthy as its database.

Check system integrity by running tripwire --check as a daily cronjob:

```
# more /etc/cron.daily/tripwire-check
#!/bin/sh
HOST_NAME=`uname -n`
if [ ! -e /var/lib/tripwire/${HOST_NAME}.twd ] ; then
     echo "****    Error: Tripwire database for ${HOST_NAME} not found.    ****"
     echo "**** Run "/etc/tripwire/twinstall.sh" and/or "tripwire --init". ****"
else
        test -f /etc/tripwire/tw.cfg &&  /usr/sbin/tripwire --check
fi
```

This cronjob automatically runs once a day, and emails the report to root. Notice that all root emails are forwarded to Biostats system group.

8. Backup

I have spent couple of days on this machine, I need a full backup to protect it. I choose the Linux bare-metal backup & recovery procedure described in UNIX Backup & Recovery by W. Curtis Preston. The procedure breaks down into six major steps:

1. Back up the important metadata.
2. Back up the operating system with a native utility.
3. Boot from alternate media.
4. Partition and format the new root disk.
5. Restore the operating system information.
6. Restore the boot block to the new root disk.

Here I only describe how I backup the important metadata and the operating system, and transfer the backup data to a save media.

Backup the metadata by running the command:

```
# fdisk -l /dev/hda > /etc/fdisk-l.txt
```

Backup the operating system using tar to a file. Remember to exclude directories /home/backup, /tmp and /proc.

```
# cat > /home/backup_exclude
home/backup
tmp
proc
^D
# mkdir /home/backup
# cd / ; tar cfX - /home/backup_exclude . | gzip -c > /home/backup/root.tar.gz
```

When it finishes, move the backup file to /home/rwang (my home directory), change permission, and I pick it up from Biostats system through SSH channel, and write to Biostats system's backup types.

Notice that I don't run daily backup, because we can always ask the research centers to resent data to us.

9. Bring the system online

Notify the Firewall administrator to allow ftp access from 12.161.125.18 and 152.73.145.21 to this machine, IP address 169.146.60.136.

Reference

Linux Practicum, by Lee Brotzman

Running UNIX Applications Securely, by Lee Brotzman and Hal Pomeranz

Securing Linux Step-By-Step, v.1.0, by the Sans Institute

Securing and Optimizing Linux: RedHat Edition, by Gerhard Mourani, (ISBN 0-9700330-0-1)

Linux System Security, The Administrator's Guide to Open Source Security Tools, by Scott Mann and Ellen L. Mitchell, (ISBN 0-13-015807-0)

Unix Backup & Recovery, by W. Curtis Preston, (ISBN 1-56592-642-0)