Securing HPUX devices.

Introduction.

In this document I am providing a checklist to be carried out on HPUX devices prior to adding them into a production environment, with the emphasis on the word "prior". Each of the items listed in my checklist seem minor on its own, but all together will help build a secure system.

There are two security policies: Open and Least Privilege. An Open security policy is where you start with full access and later disable what you consider to be a threat. With Least Privilege you turn off everything to begin with and then only turn on what is necessary to get the job done. Least Privilege is obviously the most secure option and also is the option that will cause less grief to the Systems Administration group. When you make changes on a production device, whether it is to turn off a service or change a file's permission there will always be someone who will be affected. With Least Privilege they would never have had the access to begin with.

Having correct file permissions is something vital to security. I believe that all critical root-owned files should have permissions of 400 (read-only by owner) unless absolutely necessary. If an administrator does not know how to write to these files they should probably not be accessing.

These files can be changed by either changing the permissions prior to making the change and then changing them back to the original permissions following the change, or by the use of ":w!" while editing the file using the vi editor.

Any checklist should be constantly evolving to keep up with new threats that arise. I will not go into too much detail on each of the points in the checklist. Please use my references below to get more details. My sources are both from the Internet and from current and previous work environments.

Checklist.

- ***Root accountability.***

Anyone who works in a busy work environment knows that one of the greatest threats to the integrity of a network can be a mistake made from within the company. Mis-typing a command while logged in as root to a system can cause major outages, the “rm” command being the greatest culprit. For this reason it is imperative that all root access be tracked.

Fewer people having root access on a device will cut down on the amount of mistakes that occur. However in any medium to large network there are usually a lot of people that require root access.

Add lines to root’s .profile to track who has accessed the device as root and save the shell history to a separate file for each systems administrator. Assign a three-digit access code to each systems administrator and store all these in a file called .check. When they “su –“ to root they should be prompted for this three-digit access code which will be matched against the file .check. All commands they type while logged in as root will be stored in the separate .sh_history.xxx file, where xxx is the three-digit initial.

```
grep asmith /.check
asm asmith Alan Smith 919.222.1234


tail /.sh_histories/.sh_history.asm

# Login:  asm PID:18022 Wed May 30 12:50:30 GMT 2001
cd /etc
chmod 444 shells
chmod 444 hosts.equiv
vi issue
cd /
vi .rhosts
exit
```

There are certain root-owned files that are especially sensitive. It is a good idea to track any changes to these files. A Cron job should be setup to compare the contents of these files to a saved copy every few hours. If there is a difference in file-size an email should be sent to the security mailbox. You can head off a lot of problems by having been notified promptly of changes to these important files.

```
/.profile
/.rhosts
etc/profile
etc/hosts.equiv
etc/nsswitch.conf
```

```
etc/inetd.conf
etc/ftpd/ftpusers
```

- ***Tracking user access.***

UNIX has a lot of files that will track user access to a system.
/etc/utmp contains information on all users currently logged into a system.
/var/adm/btmp contains information on all failed login attempts to a system.
/var/adm/wtmp contains all login and logouts to a system.
To ensure that these files are being written to the permissions should be set to 644.
Commands such as “w”, “last” and “who” use the above files to get their output.
The file /var/adm/sulog tracks all attempts to “su” to root by regular users.

- ***Secondary root IDs.***

Remove all ids that have user-id (UID) that makes them super user. In the example below user “lsmith” has uid of 0, same as root. This should be changed to a UID other than 0.

```
grep :0: /etc/passwd
root:h6o5acoURs6uA:0:3::/:/sbin/sh
lsmith:Zdfzxhdzhg:0:20:Larry
Smith:/export/home/lsmith:/usr/bin/ksh

grep :0: /etc/group
root::0:root
```

- ***ftpusers check.***

Check the ftpusers file to make sure it has correct entries. ftpd rejects remote logins to local user accounts that are named in /etc/ftpd/ftpusers. Each restricted account name must appear alone on a line in the file. Recommended permissions are 444.
In the example below, all listed users will not be allowed access device. All vendor supplied accounts should be included in this file.

```
more /etc/ftpd/ftpusers
root
daemon
bin
sys
adm
uucp
lp
nuucp
hpdb
nobody
admin
patrol
```

- ***Lock root login to console.***

If the /etc/securetty file is present, login security is in effect. Only user root is allowed to log in successfully on the ttys listed in this file. Restricted ttys are listed by device name, one per line. Sample entries are:

```
          console
          tty01
          ttya1
          etc
```

I recommend having only "console" as the only entry.

- ***Entries in file /etc/hosts.equiv and /.rhosts***

Make sure that /etc/hosts.equiv and /.rhosts have only needed entries. These files authorize access by remote hosts and users on local device. The use of /.rhosts is not recommended, but in certain cases it is required. Permissions of 400 are recommended for this file. Again, if someone does not know how to write to the file they should probably not be accessing the file.
For regular users make sure that no $HOME/.rhosts files exist. If there is a business need for a user to have a $HOME/.rhosts file make sure it has correct entries and the permissions are 600.

- ***umask check.***

The umask command sets the value of the file mode creation mask or displays the current one. The mask affects the initial value of the file mode (permission) bits for subsequently created files. Make sure that the following files have an appropriate umask statement:

```
/etc/profile
/.profile
/sbin/init.d/inetd
```

A secure umask value is 077, but this is not practical. A umask of 022 is recommended. Note. The umask value is in octal.

- ***No login allowed unless user has a home directory.***

If a user does not have a home directory on a device they should not be allowed to telnet or rlogin to the device. Add lines to /etc/profile to achieve this.
The code will search under /users for the username entered. If the directory does not exist then they should not be allowed on the device.

- ***Banner files at login.***

Certain files such as /etc/copyright, /etc/motd and /etc/issue should display a warning message at login. NOTE. Do **not** use the word “welcome” in any of these files. Hackers have avoided prosecution in the past because of this. When I attended the SANS GIAC class recently, an FBI agent who was attending the class gave a brief talk on this.

```
more /etc/issue
**** device **** [HPUX B.11.00]

more /etc/copyright
                            Warning Notice

This system is restricted solely to ABC Inc. authorized users for
legitimate business purposes only. The actual or attempted
unauthorized access, use, or modification of this system is
strictly prohibited . . . .

more /etc/motd

        ***********************************************
        *     Production Silly Server abc123456       *
```

****

- ***NTP***

Make sure NTP (network time protocol) is set up OK. Having all your systems synchronized with the same time can help troubleshoot break-in attempts after the fact and can help gather evidence for court cases. It is also a good idea to have all devices synchronized for the same time zone. The time zone is set in the file /etc/TIMEZONE. NTP can also be setup for higher security by configuring DES+MD5 authentication keys on server and client.
The following command will tell if you are running NTP:
/usr/sbin/ntpq -p

- ***Disable clean log.***

Make sure that the variable CLEAN_ADM is set to 0. This makes sure that the "su" log is not removed following reboot.

```
more /etc/rc.config.d/clean
#!/sbin/sh
# CLEAN_ADM:    Set to 1 to move old log files out of the way.
#               /sbin/init.d/clean_adm.
CLEAN_ADM=0
```

- ***Check that all files in root's path have correct permissions.***

Make sure that all the directories listed in root’s path are not world-writeable. A world-writeable directory would allow any user to add a program to the directory and execute it. The following piece of code will give a long listing of all directories in root’s path. Also make sure that “.” is not included in root’s path.

```
echo $PATH | tr ':' '\012' |  while read dir
do
        ls -Lld $dir
done
```

- ***Add entry to /etc/syslog.conf to log to logserver.***

Syslog is a messaging facility which allows messages to be logged to local or remote systems. Each message in the syslog.conf file is a single line of text with an associated facility and severity. Severities are hierarchical. See below for sample line in this file.

```
 tail /etc/syslog.conf
#
*.info;mail,local7.none @logserver
#
```

- ***Make sure the device is on list of systems being backed up.***

Backups are important. Any company should have a good backup plan to protect their data. One full backup per week and daily incremental backups is a good start. All backup tapes should be locked in secure cabinets. Backup tapes should be taken off-site, and these tapes should be locked up also.

- ***Make sure root's permissions in Crontab are 400.***

Cron jobs are automated processes run at scheduled times. Any Cron jobs run by root need to be secure. The permissions on root's Cron should be 400. Ensure that any scripts called in Cron have the full path spelled out. This will ensure that the intended script is run and not a Trojan Horse.

```
ls -l /usr/spool/cron/crontabs
total 2
-r--------   1 root       sys            749 Nov 30 21:46 root
```

- ***Only valid services included in /etc/inetd.conf file.***

HPUX executes many services. However, most of these services are not needed and pose a potential security risk. The first place to start is /etc/inetd.conf. This file specifies which services the /usr/sbin/inetd daemon will listen for. You eliminate unnecessary services by commenting them out. The policy of least privilege is recommended here, where all but required services are commented out of the file.

```
more /etc/inetd.conf
# A line in the configuration file has the following fields
separated by
# tabs and/or spaces:
#finger      stream tcp nowait bin  /usr/lbin/fingerd  fingerd
```

```
login         stream tcp nowait root /usr/lbin/rlogind  rlogind
telnet        stream tcp nowait root /usr/lbin/telnetd  telnetd
shell         stream tcp nowait root /usr/lbin/remshd   remshd
ftp           stream tcp nowait root /usr/lbin/ftpd      ftpd -l
```

If possible disable ftp but if it is needed use the "-l" option to enable logging.

- ***Implement TCP Wrappers***

TCP wrappers can monitor and filter incoming requests for telnet, ftp, rsh and other services running in /etc/inetd.conf.
The daemon gets run instead of the original daemon in the inetd.conf file. It does a check to see if the host is allowed to connect and then runs the original daemon if it passes the check.

- ***Disable "sendmail"***

Since the 1988 Internet Worm attack exposed "sendmail" as vulnerable, it is recommended turning off sendmail unless completely necessary. Edit the startup file:

```
more /etc/rc.config.d/mailservs
```

```
export SENDMAIL_SERVER=0
# the above may have been set to 1, change it to 0 so that
following next reboot sendmail will be disabled on the device.
```

Do not forget to stop the running sendmail process:
/sbin/init.d/sendmail stop

- ***Check for required patches.***

For UNIX devices there are security patches should be installed to combat the latest vulnerabilities. Use the command "swlist" to give all patches installed on a device (see sample below). HP keeps a listing of the latest patches on it's website.
ftp://ftp.itrc.hp.com/hp-ux patches/

```
  XR48_MAR00+SECURITY   A.1.0         HP-UX 11.00 MAR 2000
Patches
```

```
  XSWGR1100              B.11.00.48     HP-UX General Release
Patches, March 2000
```

- ***Change root's home directory.***

The default home directory for root is /. It is a good idea to change this to /root. The reason for this is to give root it's own home directory. Often files are mistakenly left in the / directory with less than safe permissions.
The /etc/passwd file entry for root would look something like:

```
     root:*:0:3::/root:/sbin/sh
```

- ***Tftp secure***

If tftp is not needed it should be removed from all files such as /etc/inetd.conf and /etc/passwd. If it is required then the entry in /etc/passwd should look like:

```
tftp:*:201:20:,,,:/users/tftpdir:/bin/false
```

The "*" in the second column blocks the account and the last column sets the shell to /bin/false, which would not allow anyone to log in using the account.

Conclusion.

I hope that the checklist I have given can help to improve the security policies of anyone reading this. Networking is a changing world and along with this defense policies need to change. To keep up with new vulnerabilities get added to mailing lists that send out alerts (such as SANS) and take the appropriate measures.

References:

http://www.spy.net/~jeeb/unix_security.html

http://people.hp.se/stevesk/bastion11.html

http://www.cert.org/tech_tips

http://www.unixtools.com/security.html

http://www.deter.com/unix/papers/cert_generic.txt

UNIX system security: A Guide for Users and System Administrators
By David A. Curry
Addison-Wesley Professional Computing Series