building a bastion host using hp-ux 11

august 2000

a white paper from the Hewlett-Packard Company

table of contents

abstract

A bastion host is a computer system that is exposed to attack, and which also may be a critical component in a network security system. Special attention must be paid to these highly fortified hosts, both during initial construction and ongoing operation. Bastion hosts can include:

- Firewall gateways
- Web servers
- FTP servers
- Name servers (DNS)
- Mail hubs
- Victim hosts (sacrificial lambs)

This paper presents a methodology for building a bastion host using HP-UX 11, and walks through the steps used to build a sample, generic bastion host using HP-UX 11.00. While the principles and procedures can be applied to other HP-UX versions, as well as other UNIX® variants, our focus is on HP-UX 11.

what is a bastion host?

The American Heritage Dictionary defines a bastion as:

1. A projecting part of a rampart or other fortification. 2. A well-fortified position or area. 3. Something regarded as a defensive stronghold.

Marcus Ranum[1] is generally credited with applying the term bastion to hosts that are exposed to attack, and its common use in the firewall community. He says:

> *Bastions are the highly fortified parts of a medieval castle; points that overlook critical areas of defense, usually having stronger walls, room for extra troops, and the occasional useful tub of boiling hot oil for discouraging attackers. A bastion host is a system identified by the firewall administrator as a critical strong point in the network's security. Generally, bastion hosts will have some degree of extra attention paid to their security, may undergo regular audits, and may have modified software.*

Bastion hosts are not general purpose computing resources. They differ in both their purpose and their specific configuration. A victim host may permit network logins so users can run untrusted services, while a firewall gateway may only permit logins at the system console. The process of configuring or constructing a bastion host is often referred to as hardening.

The effectiveness of a specific bastion host configuration can usually be judged by answering the following questions:

- How does the bastion host protect itself from attack?
- How does the bastion host protect the network behind it from attack?

Extreme caution should be exercised when installing new software on bastion hosts. Very few software products have been designed and tested to run on these exposed systems.

[1] Marcus J. Ranum, "Thinking About Firewalls," SANS 1993.

methodology for building a bastion host

Let's begin by creating a methodology. These are the principles and procedures we will follow as we build bastion hosts. Included in this is our mindset, which will help guide the configuration decisions we make. And we want our mindset to be paranoid.

We start with a clean operating system install. If subsystems are not needed for the applications we plan to run on the bastion host, we will not install them in the first place, or disable or remove them after the install.

Next we install any additional operating system software needed on the bastion host, such as network drivers not available on the install media or the LVM Mirror product, followed by the latest patch bundle (Support Plus Bundle). We perform a security patch review and install HP-UX security patches that apply to our installed software configuration. The system is configured with commercial security (as a trusted system) which removes the hashed passwords from the `/etc/passwd` file and provides other useful security features such as auditing and login passwords with lengths greater than 8 characters. Unneeded pseudo-accounts in the password database are removed.

We remove the set-id bits from all programs then selectively add them back to programs that must be run by non-privileged users. This proactive approach may save us time and a future vulnerability window when the next security defect is discovered in a set-id program.

We tighten up the world-write permissions on system files, and set the sticky bit on publicly writable directories. We next set a number of tunable network parameters, taking a paranoid stance toward security. At this point, the applications that will run on the bastion host can be installed, configured and tested. This may include installing additional security software, such as TCP wrappers and SSH. After testing is complete, we create a bootable System Recovery Tape of the root volume group.

sample blueprint for a bastion host

Now let's lay out the blueprint that we'll use as we construct a sample, generic bastion host using HP-UX 11.00:

1. Install HP-UX
2. Install additional products
3. Install Support Plus bundle
4. Install security patches
5. Take first steps toward security
6. Disable network services
7. Disable other daemons
8. Examine set-id programs
9. Examine file permissions
10. Perform security network tuning
11. Install software and test configuration
12. Create system recovery tape

Keep in mind that this is a sample starting configuration, and you will need to make changes specific to your planned use of the system. If you're installing a future HP-UX version such as HP-UX 11i, some things may be different. You may also choose to reorder things slightly for various reasons. Every bastion host is different.

Document your configuration steps as you perform them—you may discover later that a change that was made causes unforeseen problems. And it may take several installations to get everything working correctly.

1. install hp-ux

It takes at most one hour to install a minimal HP-UX configuration from CD-ROM. The security benefits of starting with a clean operating system install, and knowing exactly what we have, far exceed this minor cost in time. Even if our host is new and has been shipped from the factory with HP-UX preinstalled, we should reinstall from scratch.

During the initial installation, configuration and testing, we must make sure that our system is not connected to any untrusted networks. We may want to connect the system to a network only after we have completed our configuration steps.

The example used for this paper employs a completely private network (e.g., hub or cross-cable) connected only to the LAN console. Note that the test system used is an HP L2000, which will only run 64-bit HP-UX; we are also using the 9911 install media (11.ACE).

To perform the installation we boot from the install CD and perform the following steps:

1. Select “Install HP-UX”
2. In the “User Interface and Media Options” screen select:
   a. Media-only installation
   b. Advanced Installation
3. In the “Basic” screen select Environments “64-Bit Minimal HP-UX (English Only)”
4. In the “Software” screen:
   a. Select “Change Depot Location”
   b. Change “Interactive swinstall” to “Yes”
   c. Select “Modify”
5. Change other configuration settings as appropriate for your system
6. Select “Go!”
7. In the “SD Install” screen:
   a. Change the Software View to Products:
      View->Change Software View->Start with Products
   b. Mark MailUtilities.Runtime and MailUtilities.Manuals for Install
   c. Unmark NFS.Runtime.NIS-CLIENT for Install. (This will also unmark KEY-CORE and NIS-CORE.)
   d. Unmark NFS.Runtime.NFS-CLIENT for Install.
   e. Mark NFS.Runtime.NFS-64SLIB for Install.
   f. Unmark Networking.MinimumRuntime.PPP-RUN for Install.
   g. Select OS-Core.Manuals for Install.
   h. Select SOE for Install.
   i. Select SecurityMon for Install.
   j. Select Streams.Runtime.STREAMS-64SLIB for Install.
   k. Select SystemAdmin.Runtime for Install.
   l. Select TextEditors.Runtime and TextEditors.Manuals for Install.
   m. Perform installation analysis:
      Actions->Install (analysis)

We choose a minimal HP-UX system. This will not install the X-window system and many other products that we don’t need or want.

We remove as much of the NFS product as possible, because it has a number of security problems and we will not be using it. We also remove the PPP-RUN fileset because we are not using PPP.

For system management purposes we install SAM, the core OS man pages, mailers and text editors. We will be using the commercial security feature of HP-UX, so we need to select the SecurityMon and SOE products. (SecurityMon contains commands and documentation for auditing and trusted system components, and SOE contains the ***pwconv*** command which we will use below.) Finally, since we are installing on 64-bit hardware, we select the 64-bit libraries for NFS and STREAMS, which are required for various applications.

We would like to remove other products such as SNMP (OVSNMPAgent). In the case of SNMP, however, a number of other products are dependent upon it). So we disable SNMP and other products that are difficult or impossible to remove.

This procedure yields a relatively lean configuration. Much of the space in */var/* is for saved patches, which we can optionally remove later. The following output of ***bdf***, ***ps-ef*** and ***netstat -anf inet*** illustrates just how lean the configuration is:

```
# uname -a
HP-UX bastion B.11.00 A 9000/800 137901517 two-user license

# bdf
Filesystem          kbytes    used   avail %used Mounted on
/dev/vg00/lvol3     143360   18699  116899   14% /
/dev/vg00/lvol1      83733   15965   59394   21% /stand
/dev/vg00/lvol8     512000  123680  364879   25% /var
/dev/vg00/lvol7     512000  164352  325949   34% /usr
/dev/vg00/lvol4      65536    1122   60394    2% /tmp
/dev/vg00/lvol6     262144    3513  242523    1% /opt
/dev/vg00/lvol5      20480    1109   18168    6% /home

# ps -ef
   UID   PID  PPID  C    STIME TTY       TIME COMMAND
  root     0     0  0 14:21:25 ?         0:10 swapper
  root     1     0  0 14:21:25 ?         0:00 init
  root     2     0  0 14:21:25 ?         0:00 vhand
  root     3     0  0 14:21:25 ?         0:00 statdaemon
  root     4     0  0 14:21:25 ?         0:00 unhashdaemon
  root     8     0  0 14:21:25 ?         0:00 supsched
  root     9     0  0 14:21:25 ?         0:00 strmem
  root    10     0  0 14:21:25 ?         0:00 strweld
  root    11     0  0 14:21:25 ?         0:00 strfreebd
  root    12     0  0 14:21:25 ?         0:00 ttisr
  root    18     0  0 14:21:25 ?         0:00 lvmkd
  root    19     0  0 14:21:25 ?         0:00 lvmkd
  root    20     0  0 14:21:25 ?         0:00 lvmkd
  root    21     0  0 14:21:25 ?         0:00 lvmkd
  root    22     0  0 14:21:25 ?         0:00 lvmkd
  root    23     0  0 14:21:25 ?         0:00 lvmkd
  root   826     1  0 14:25:12 console   0:00 -sh
  root   522     1  0 14:24:48 ?         0:00 /usr/sbin/ptydaemon
  root   870   866  1 14:30:26 console   0:00 ps -ef
  root    28     0  0 14:21:26 ?         0:00 vxfsd
```

```
  root   460     1  0 14:24:46 ?         0:00 /usr/sbin/syncer
  root   708     1  0 14:24:58 ?         0:00 /usr/sbin/snmpdm
  root   651     1  0 14:24:57 ?         0:00 /usr/sbin/rpcbind
  root   519     1  0 14:24:48 ?         0:00 /usr/sbin/syslogd -D
  root   535     1  0 14:24:49 ?         0:00 /usr/lbin/nktl_daemon 0 0 0 0 0 1 -2
  root   656     0  0 14:24:57 ?         0:00 nfskd
  root   545     1  0 14:24:52 ?         0:00 /usr/lbin/ntl_reader 0 1 1 1 1000 /var/adm/nettl /var/adm/co
  root   546   545  0 14:24:52 ?         0:00 /usr/sbin/netfmt -C -F -f /var/adm/nettl.LOG00 -c /var/adm/c
  root   746     1  0 14:25:09 ?         0:00 /usr/sbin/cron
  root   680     1  0 14:24:57 ?         0:00 /usr/sbin/inetd
  root   703     1  0 14:24:58 ?         0:00 sendmail: accepting connections on port 25
  root   866   826  0 14:28:53 console   0:00 ksh
  root   719     1  0 14:25:08 ?         0:00 /usr/sbin/hp_unixagt
  root   727     1  0 14:25:09 ?         0:06 /usr/sbin/mib2agt
  root   735     1  0 14:25:09 ?         0:00 /usr/sbin/trapdestagt
  root   743     1  0 14:25:09 ?         0:00 /usr/sbin/pwgrd
  root   749     1  0 14:25:09 ?         0:00 /usr/sbin/envd
  root   758     1  0 14:25:09 ?         0:00 /usr/sbin/swagentd -r
```

```
# netstat -anf inet
Active Internet connections (including servers)
Proto Recv-Q Send-Q  Local Address          Foreign Address         (state)
tcp        0      0  *.7161                 *.*                     LISTEN
tcp        0      0  *.544                  *.*                     LISTEN
tcp        0      0  *.543                  *.*                     LISTEN
tcp        0      0  *.515                  *.*                     LISTEN
tcp        0      0  *.514                  *.*                     LISTEN
tcp        0      0  *.513                  *.*                     LISTEN
tcp        0      0  *.512                  *.*                     LISTEN
tcp        0      0  *.113                  *.*                     LISTEN
tcp        0      0  *.111                  *.*                     LISTEN
tcp        0      0  *.37                   *.*                     LISTEN
tcp        0      0  *.25                   *.*                     LISTEN
tcp        0      0  *.23                   *.*                     LISTEN
tcp        0      0  *.21                   *.*                     LISTEN
tcp        0      0  *.19                   *.*                     LISTEN
tcp        0      0  *.13                   *.*                     LISTEN
tcp        0      0  *.9                    *.*                     LISTEN
tcp        0      0  *.7                    *.*                     LISTEN
udp        0      0  *.2121                 *.*
udp        0      0  *.514                  *.*
udp        0      0  *.111                  *.*
udp        0      0  *.*                    *.*
udp        0      0  *.49152                *.*
udp        0      0  *.518                  *.*
udp        0      0  *.13                   *.*
udp        0      0  *.7                    *.*
udp        0      0  *.9                    *.*
udp        0      0  *.19                   *.*
udp        0      0  *.161                  *.*
udp        0      0  *.*                    *.*
udp        0      0  *.*                    *.*
udp        0      0  *.*                    *.*
```

Even though the configuration is lean, we still have work to do.

2. install additional products

At this point, we install any additional HP products that are required on the bastion host—for example, network drivers for add-on LAN cards, or other products we plan to use, such as LVM Mirror. We install a portion of the HP Ignite product to obtain the software (***make_recovery*** command) required to build a bootable backup tape of the root volume group, which we will create at the end of the configuration process.

For our sample configuration, we are using the 4-Port 100BT PCI card, so we need to install the driver for that card. We will also install the required filesets in Ignite-UX for ***make_recovery*** functionality.

Using the December 1999 Applications CD as an example, we install the following product and filesets:

- 100BASE-T
- Ignite-UX.BOOT-KERNEL
- Ignite-UX.FILE-SRV-11-00
- Ignite-UX.MGMT-TOOLS
- Ignite-UX.RECOVERY

3. install the support plus bundle

Next we install all General Release (GR) patches from the latest HP-UX 11.0 Support Plus CD, which in the example is from December 1999. The install CD contained a recent set of patches from the time the media was produced (November 1999) so in this case we don't expect to have many patches selected. We mount the Support Plus CD and use ***swinstall*** to install the GR bundle XSWGR1100.

4. install security patches

We next perform a security patch review to determine if any security patches should be installed. HP-UX patches are available via anonymous FTP[2]. Note that due to the unification of install media and the kernel with 11.00, all 11.X patches are currently contained in */hp-ux_patches/s700_800/*. However there may be future platform-specific patches for s700 and s800.

An HP-UX Patch Security Matrix[3] is also available, which contains a list of current security patches for each HP-UX platform and operating system version combination (for example, s800 11.00). The matrix is updated nightly. There is also a list of the MD5 hash codes[4] for each patch, which can be used to verify that patches we intend to install have not been tampered with.

For our sample s800, 11.00 host, the current security patches at the time of this writing are:

```
s800 11.00:PHCO_19945 s700_800 11.00 bdf(1M) patch to skip autofs file systems
           PHCO_20078 s700_800 11.0 Software Distributor (SD-UX) Cumulative Patch
           PHCO_20765 s700_800 11.00 libc cumulative patch
           PHKL_20315 s700_800 11.00 Cumulative LOFS patch
           PHNE_16295 s700_800 11.00 vacation patch.
           PHNE_17028 s700_800 11.00 r-commands cumulative mega-patch
           PHNE_17190 s700_800 11.00 sendmail(1m) 8.8.6 patch
           PHNE_17949 s700_800 11.00 Domain Management (DESMS B.01.12)
           PHNE_18017 s700_800 11.00 Domain Management (DESMS-NS B.01.11)
           PHNE_18377 s700_800 11.00 ftpd(1M) and ftp(1) patch
           PHNE_19620 s700_800 11.0 ONC cumulative patch
           PHNE_20619 s700_800 11.00 Bind 4.9.7 components
           PHNE_20735 s700_800 11.00 cumulative ARPA Transport patch
           PHSS_16649 s700_800 11.00 Receiver Services October 1998 Patch
           PHSS_17310 s700_800 11.00 OV OB2.55 patch - WinNT packet
           PHSS_17483 s700_800 11.00 MC/LockManager A.11.05 (English) Patch
           PHSS_17484 s700_800 11.00 MC/LockManager A.11.05 (Japanese) Patch
           PHSS_17496 s700_800 11.00 Predictive C.11.0[0,a-m] cumulative patch
           PHSS_17581 s700_800 11.00 MC ServiceGuard 11.05 Cumulative Patch
           PHSS_20385 s700_800 11.00 OV OB2.55 patch - DA packet
           PHSS_20544 s700_800 11.00 OV EMANATE14.2 Agent Consolidated Patch
           PHSS_20716 s700_800 11.00 CDE Runtime DEC99 Periodic Patch
```

[2] HP-UX patches are available via anonymous FTP in North America at ftp://us-ffs.external.hp.com/hp-ux_patches/; and Europe at ftp://europe-ffs.external.hp.com/hp-ux_patches/.

[3] HP-UX Patch Security Matrix, ftp://europe-ffs.external.hp.com/export/patches/hp-ux_patch_matrix.

[4] HP-UX Patch Checksum Information, ftp://europe-ffs.external.hp.com/export/patches/hp-ux_patch_sums.

Each patch for a product currently installed on the system should be analyzed to determine if it needs to be installed. First we check and see if it's already installed from either the install media or the patch bundle. If not, we can look at the *patch.text* file for details about the patch, including dependencies, filesets affected, and files patched. We can determine filesets installed on the system by executing ***swlist -l fileset***.

Just because a patch exists doesn't mean that we need to install it, though it is safest to do so. Some patches may fix buffer overrun defects or other attack channels in set-uid root commands or root processes. If we plan to remove the set-uid bits, we may simply choose not to install them. We may also not have a program configured (for example, *rlogind* listening on the network), but sometimes it can be difficult to determine if a defect is remotely or locally exploitable. If we're not sure whether a particular patch needs to be installed, it's best to just install it.

We also examine the security bulletins themselves[5], because not all security bulletins result in a patch. For example, there is a security bulletin regarding the default PMTU strategy that recommends its default be changed using ***ndd*** (HPSBUX0001-110). Another bulletin highlights a serious issue with blank password fields when using Ignite-UX and trusted systems (HPSBUX0002-111). (We will address the issue with the PMTU setting below when we set network security tunables; the Ignite-UX issue concerns ***make_sys_image***, which we will not be using.)

5. take the first steps

There are a few miscellaneous configuration and cleanup steps we can perform immediately after the operating system install and patch steps.

remove saved patches (optional)

By default during patch installation, rollback copies of all patch files modified are saved in */var/adm/sw/save/*. We may wish to remove these files and claim the disk space by marking the patches "committed." (Note, however, that if we do this, there will be no way to uninstall the patch with ***swremove***.) To remove the patches following a fresh install, execute the following command:

```
# swmodify -x patch_commit=true '*.*'
```

convert to a trusted system

We use the tsconvert command to convert to a trusted system:

```
# /usr/lbin/tsconvert
Creating secure password database...
Directories created.
Making default files.
System default file created...
Terminal default file created...
Device assignment file created...
Moving passwords...
secure password database installed.
Converting at and crontab jobs...
At and crontab files converted.
# passwd root
```

Passwords on existing accounts will expire as a result of the conversion, which is why we change the root password. We may also want to enable auditing.

[5] HP Security Bulletins are available at http://us-support.external.hp.com/ and http://europe-support.external.hp.com/. Select "Search Technical Knowledge Base." You need a login to access security bulletins, but you can register for one in a few minutes.

tighten global privileges

HP-UX has a feature known as privilege groups, which is a mechanism to assign privileges to a group (see ***privgrp(4)***). By default the ***chown*** privilege is a global privilege and applies to all groups:

```
$ getprivgrp
global privileges: CHOWN
```

Non-privileged users really don't need to be able to ***chown*** files to other users. (In Linux, for example, only the super-user can change the owner of a file.) */sbin/init.d/set_prvgrp* is executed by default at system startup and executes the command ***/usr/sbin/setprivgrp -f /etc/privgroup*** if */etc/privgroup* exists. We can create a configuration file that will delete all privileges for all groups (see ***setprivgrp(1m)***):

```
# getprivgrp
global privileges: CHOWN
# echo -n >/etc/privgroup
# chmod 400 /etc/privgroup
# /sbin/init.d/set_prvgrp start
# getprivgrp
global privileges:
```

fix PAM CDE problems

SAM will perform some correctness checks on */etc/pam.conf* that involve trying to find a command using several different paths for each *service_name*. We did not install CDE and yet our *pam.conf* file contains *dtlogin* and *dtaction* entries for each of the PAM module types; for example:

```
dtlogin  auth required  /usr/lib/security/libpam_unix.1
dtaction auth required  /usr/lib/security/libpam_unix.1
```

We can safely remove these, which will permit us to access the authenticated command's functionality in SAM:

```
# cp /etc/pam.conf /etc/pam.conf.SAVE
# grep -Ev '^(dtlogin|dtaction)' /etc/pam.conf.SAVE >/etc/pam.conf
```

fix hparray startup symlinks

There are some startup symlinks pointing to array startup scripts that are contained in filesets that we do not have and do not need (OS-Core.C2400-UTIL and OS-Core.ARRAY-MGMT). So we remove them:

```
# for f in /sbin/rc*.d/*; do [ ! -f $f ] && echo $f; done
/sbin/rc1.d/K290hparamgr
/sbin/rc1.d/K290hparray
/sbin/rc2.d/S710hparamgr
/sbin/rc2.d/S710hparray
# rm /sbin/rc1.d/K290hparamgr
# rm /sbin/rc1.d/K290hparray
# rm /sbin/rc2.d/S710hparamgr
# rm /sbin/rc2.d/S710hparray
```

set the default umask

One side effect of converting to a trusted system is that the default umask of 0 is changed to 07077, so nothing needs to be performed to tighten up the umask.

restrict root login to the console

If desired, we can restrict the root login to the console:

```
# echo console > /etc/securetty
# chmod 400 /etc/securetty
```

enable inetd logging

If ***inetd*** will remain enabled, we next add the *-l* (minus ell) argument to the *INETD_ARGS* environment variable in */etc/rc.config.d/netdaemons*:

```
export INETD_ARGS=-l
```

remove unneeded pseudo-accounts

To remove unneeded pseudo-accounts, we first examine some groups that might be removed, then we examine users. The basic strategy is that if there are no processes that are run with a given user or group, and there are no files owned by a user or group, we remove them:

```
# find / -group lp -o -group nuucp -o -group daemon -exec ls -ld {} \;
# groupdel lp
# groupdel nuucp
# groupdel daemon
# find / -user uucp -o -user lp -o -user nuucp -o -user hpdb \
> -o -user www -o -user daemon -exec ls -ld {} \;
# userdel uucp
# userdel lp
# userdel nuucp
# userdel hpdb
# userdel www
# userdel daemon
```

For the remaining pseudo-accounts (bin, sys and adm), we change the login shell to some invalid path; for example /. Or we use the ***noshell*** program from the Titan package[6].

```
# pwget -n bin
bin:*:2:2:NO LOGIN:/usr/bin:/
```

configure nsswitch.conf(4) policy

If we are going to configure the DNS resolver, we do it at this point. Many bastion hosts, including firewall gateways, do not have DNS configured at all. For these hosts, we set the ***nsswitch.conf(4)*** to search local files only:

```
# cp /etc/nsswitch.files /etc/nsswitch.conf
# chmod 444 /etc/nsswitch.conf
```

change root home directory to /root

We change root's home directory from the default of */* to */root*. Our motivation is to give the root account a private home directory to lessen the possibility of files being placed unintentionally in */*. This also permits us to put a restrictive mode on the directory. We edit */etc/passwd* and change root's entry to the following:

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

Then we build the directory and update the TCB:

```
# mkdir /root
# chmod 700 /root
# mv /.profile /root
# pwconv
Updating the tcb to match /etc/passwd, if needed.
```

[6] Titan host security tool, http://www.fish.com/titan/.

6. disable network services

Our next step in creating a bastion host is to disable network services.

disable inetd services

We should be able to identify each TCP and UDP service emitted by ***`netstat -af inet`***. Those services that are not needed or cannot be secured should be disabled. Examples of such services include the UDP and TCP small servers, such as echo, chargen, daytime, time and discard; the Berkeley r* services; talk, etc.

Some bastion hosts have an entirely empty *`inetd.conf`*. We can start by removing all services from *`inetd.conf`*, restarting it, then examining the ***`netstat`*** output. If we stick with a bare *`inetd.conf`*, we can choose to not run ***`inetd`*** at all. To disable ***`inetd`*** startup and shutdown, we remove the corresponding symbolic links from the *`rc`* directories:

```
# rm /sbin/rc2.d/S500inetd
# rm /sbin/rc1.d/K500inetd
```

For the remaining services, we may want to use ***`inetd.sec(4)`***, which permits IP-address-based authentication of remote systems.

With all services removed from the *`inetd.conf`* file, ***`netstat`*** yields the following:

```
# netstat -af inet
Active Internet connections (including servers)
Proto Recv-Q Send-Q  Local Address          Foreign Address        (state)
tcp        0      0  *.7161                 *.*                    LISTEN
tcp        0      0  *.portmap              *.*                    LISTEN
tcp        0      0  *.smtp                 *.*                    LISTEN
udp        0      0  *.2121                 *.*
udp        0      0  *.syslog               *.*
udp        0      0  *.portmap              *.*
udp        0      0  *.*                    *.*
udp        0      0  *.49152                *.*
udp        0      0  *.*                    *.*
udp        0      0  *.snmp                 *.*
udp        0      0  *.*                    *.*
udp        0      0  *.*                    *.*
```

This is much better, though we still need to determine what the remaining services are. We see that servers are listening on the UDP SNMP, portmap and syslog ports, as well as the SMTP and TCP portmap ports. However, 2121/udp, 2121/tcp, 7161/tcp and 49152/udp were not found in */etc/services*, so ***`netstat`*** is unable to print the service name. There are also some wildcard (*.*) local UDP listeners that are a mystery.

An extremely useful tool for identifying network services is ***lsof*** (LiSt Open Files)[7]. The command ***lsof -i*** shows us the processes that are listening on the remaining ports:

```
# lsof -i
COMMAND   PID USER   FD   TYPE      DEVICE SIZE/OFF NODE NAME
syslogd   261 root    5u  inet 0x10191e868      0t0  UDP *:syslog (Idle)
rpcbind   345 root    4u  inet     72,0x73      0t0  UDP *:portmap (Idle)
rpcbind   345 root    6u  inet     72,0x73      0t0  UDP *:49158 (Idle)
rpcbind   345 root    7u  inet     72,0x72      0t0  TCP *:portmap (LISTEN)
sendmail: 397 root    5u  inet 0x10222b668      0t0  TCP *:smtp (LISTEN)
snmpdm    402 root    3u  inet 0x10221a268      0t0  TCP *:7161 (LISTEN)
snmpdm    402 root    5u  inet 0x10222a268      0t0  UDP *:snmp (Idle)
snmpdm    402 root    6u  inet 0x10221f868      0t0  UDP *:* (Unbound)
mib2agt   421 root    0u  inet 0x10223e868      0t0  UDP *:* (Unbound)
swagentd  453 root    6u  inet 0x1019d3268      0t0  UDP *:2121 (Idle)
```

We see that *rpcbind* is listening on 49158/udp (it's unclear whether this is a fixed or ephemeral port assignment) and *snmpdm* is listening on 7161/tcp. Also, we see that *snmpdm* and *mib2agt* are the source of the mysterious unbound wildcard ports.

disable other services

With this information, we can proceed with the following steps.

prevent syslogd from listening on the network

PHCO_21023 can be installed, which adds the ***-N*** option to ***syslogd*** to prevent it from listening on the network for remote log messages. After installing this patch, we edit */sbin/init.d/syslogd* and modify the line that starts *syslogd* to be */usr/sbin/syslogd -DN*.

disable SNMP daemons

Next we edit SNMP startup configuration files:

`/etc/rc.config.d/SnmpHpunix`

- We setSNMP_HPUNIX_START to 0: *SNMP_HPUNIX_START=0*.

`/etc/rc.config.d/SnmpMaster`

- We setSNMP_MASTER_START to 0: *SNMP_MASTER_START=0*.

`/etc/rc.config.d/SnmpMib2`

- We setSNMP_MIB2_START to 0: *SNMP_MIB2_START=0*.

`/etc/rc.config.d/SnmpTrpDst`

- We setSNMP_TRAPDEST_START to 0: *SNMP_TRAPDEST_START=0*.

[7] Vic Abell's lsof (LiSt Open Files), ftp://vic.cc.purdue.edu/pub/tools/unix/lsof/.

disable the swagentd (SD-UX) daemon

This is complicated. The *swagentd* script is run twice in the bootup start sequence, and performs different tasks based upon its program name argument. (For example, if run as ***S100swagentd*** it will remove the files listed in */var/adm/sw/cleanupfile*.) Also, for the *swconfig* script to work properly, ***swagentd*** must be running. Our solution is to create a new script, which will be configured to run immediately after ***S120swconfig*** to kill the ***swagentd*** daemon in a paranoid fashion, and remove the other start and kill *rc* links.

The key portion of the kill script, *swagentdk*[8], follows:

```
start)
        /usr/sbin/swagentd -k
        sleep 1
        findproc swagentd
        if [ "$pid" != "" ]; then
                kill $pid
                sleep 5
                findproc swagentd
                if [ "$pid" != "" ]; then
                        kill -9 $pid
                        sleep 5
                        findproc swagentd
                        if [ "$pid" != "" ]; then
                                echo "UNABLE TO KILL SWAGENTD PROCESS!!!"
                                rval=3  # REBOOT!!!
                        fi
                else
                        rval=0
                fi
        else
                rval=0
        fi
        ;;
```

We try to kill the daemon three times, using increasing levels of force. If we can't stop the daemon using *kill -9*, we set *rval=3*, which will cause a reboot. (This drastic step may exceed your specific security and paranoia requirements.)

To configure, we perform the following:

```
# cp /tmp/swagentdk /sbin/init.d
# chmod 555 /sbin/init.d/swagentdk
# ln -s /sbin/init.d/swagentdk /sbin/rc2.d/S121swagentdk
# rm /sbin/rc2.d/S870swagentd
# rm /sbin/rc1.d/K900swagentd
```

[8] swagentdk script, http://people.hp.se/stevesk/swagentdk.

disable the sendmail daemon

We set the SENDMAIL_SERVER environment variable to 0 in */etc/rc.config.d/mailservs*:

```
export SENDMAIL_SERVER=0
```

disable the rpcbind daemon

We don't plan to run any RPC services on the bastion host and need to disable the startup of ***rpcbind*** (this is the portmap replacement on HP-UX 11.0). After some grepping in */etc/rc.config.d* we find that ***rpcbind*** is started from the *nfs.core* script, so we disable it in the *rc* startup directories. We also move the *rpcbind* program to a new name as an additional safety measure. (However, a patch install could reinstall it. So it's important to reexamine our configuration any time patches are installed on the bastion host).

```
# rm /sbin/rc1.d/K600nfs.core
# rm /sbin/rc2.d/S400nfs.core
# mv /usr/sbin/rpcbind /usr/sbin/rpcbind.DISABLE
```

This also avoids the startup of the ***nfskd*** process, which we saw in previous ***ps*** output.

After a reboot to verify the modifications made to the startup scripts, we can check the ***netstat*** and ***lsof*** output and verify that no network services remain enabled. We can also check the ***ps*** output again to verify that the disabled daemons were not launched:

```
# netstat -af inet
Active Internet connections (including servers)
Proto Recv-Q Send-Q  Local Address          Foreign Address        (state)
udp        0      0  *.*

# lsof -i
# ps -ef
     UID   PID  PPID  C    STIME TTY       TIME COMMAND
    root     0     0  0 15:59:18 ?         0:10 swapper
    root     1     0  0 15:59:19 ?         0:00 init
    root     2     0  0 15:59:18 ?         0:00 vhand
    root     3     0  0 15:59:18 ?         0:00 statdaemon
    root     4     0  0 15:59:18 ?         0:00 unhashdaemon
    root     8     0  0 15:59:18 ?         0:00 supsched
    root     9     0  0 15:59:18 ?         0:00 strmem
    root    10     0  0 15:59:18 ?         0:00 strweld
    root    11     0  0 15:59:18 ?         0:00 strfreebd
    root    12     0  0 15:59:18 ?         0:00 ttisr
    root    18     0  0 15:59:19 ?         0:00 lvmkd
    root    19     0  0 15:59:19 ?         0:00 lvmkd
    root    20     0  0 15:59:19 ?         0:00 lvmkd
    root    21     0  0 15:59:19 ?         0:00 lvmkd
    root    22     0  0 15:59:19 ?         0:00 lvmkd
    root    23     0  0 15:59:19 ?         0:00 lvmkd
    root   367     1  0 15:59:48 console   0:00 -sh
    root   206     1  0 15:59:38 ?         0:00 /usr/sbin/syncer
    root   324     1  0 15:59:47 ?         0:00 /usr/sbin/inetd -l
    root    28     0  0 15:59:20 ?         0:00 vxfsd
    root   237     1  0 15:59:39 ?         0:00 /usr/sbin/ptydaemon
```

```
root   380   367  0 16:00:03 console   0:00 ksh
root   410   380  1 16:04:05 console   0:00 ps -ef
root   250     1  0 15:59:40 ?         0:00 /usr/lbin/nktl_daemon 0 0 0 0 0 1 -2
root   356     1  0 15:59:47 ?         0:00 /usr/sbin/cron
root   260     1  0 15:59:42 ?         0:00 /usr/lbin/ntl_reader 0 1 1 1 1000 /var/adm/nettl /var/adm/co
root   261   260  0 15:59:42 ?         0:00 /usr/sbin/netfmt -C -F -f /var/adm/nettl.LOG00 -c /var/adm/c
root   352     1  0 15:59:47 ?         0:00 /usr/sbin/pwgrd
root   359     1  0 15:59:47 ?         0:00 /usr/sbin/envd
root   400     1  0 16:02:04 ?         0:00 /usr/sbin/syslogd -DN
```

Note that in this case, ***netstat*** shows a wildcard UDP listener, but ***lsof*** is silent on this. What is happening here is that ***netstat -a*** is displaying information for all open UDP STREAMS, including STREAMS that are not bound. The line above represents an unbound UDP STREAM, where a *udp_open()* occurred that was never followed by a *udp_bind()*. This is basically an orphaned UDP STREAM, which cannot send or receive because there is no path for the data to travel to and from the IP layer. An alternative and more accurate check for listening UDP endpoints is:

```
$ ndd -get /dev/udp ip_udp_status
UDP ipc        hidx lport fport laddr         faddr         flags    dist head
```

This command displays no endpoints.

7. disable other daemons

We can now examine the current process listing and determine if there are other daemons that can be disabled. Our approach is: if we aren't using it, disable it.

Many of the processes remaining are system processes. System processes can be identified by examining the flags column in a long process listing (***ps -el***). The flags field is an additive octal bit-field, like the UNIX mode bits on files. (See the ***ps*** command for a listing of the process flag bits.) The processes that have the 2 flag bit set (for example, 1003, 01000 + 2 + 1) are system processes and can probably be ignored safely. (The 01000 bit is explained on the next page.)

```
# ps -el
  F S        UID   PID  PPID  C PRI NI     ADDR   SZ             WCHAN TTY       TIME COMD
1003 S          0     0     0  0 128 20           6a4f58    0                 - ?         0:10 swapper
141  S          0     1     0  0 168 20        101d3e600  100 400003ffffff0000 ?         0:00 init
1003 S          0     2     0  0 128 20        101b25f00    0           747e90 ?         0:00 vhand
1003 S          0     3     0  0 128 20        101b36200    0           5f2060 ?         0:00 statdaemon
1003 S          0     4     0  0 128 20        101b36500    0           6ec250 ?         0:00 unhashdaemon
1003 S          0     8     0  0 100 20        101b25300    0           72fed8 ?         0:00 supsched
1003 S          0     9     0  0 100 20        101b25600    0           6a3698 ?         0:00 strmem
1003 S          0    10     0  0 100 20        101b25900    0           6f2988 ?         0:00 strweld
1003 S          0    11     0  0 100 20        101b25c00    0           6cc2d0 ?         0:00 strfreebd
1003 S          0    12     0  0 -32 20        101b36800    0           6a0c68 ?         0:00 ttisr
1003 S          0    18     0  0 147 20        101b4c000    0           6a2fb0 ?         0:00 lvmkd
1003 S          0    19     0  0 147 20        101b4c300    0           6a2fb0 ?         0:00 lvmkd
1003 S          0    20     0  0 147 20        101b4c600    0           6a2fb0 ?         0:00 lvmkd
1003 S          0    21     0  0 147 20        101b4c900    0           6a2fb0 ?         0:00 lvmkd
1003 S          0    22     0  0 147 20        101b4cc00    0           6a2fb0 ?         0:00 lvmkd
1003 S          0    23     0  0 147 20        101b4cf00    0           6a2fb0 ?         0:00 lvmkd
  1 S           0   367     1  0 158 20        101e56100  106          31fff00 console   0:00 sh
  1 S           0   206     1  0 154 20        101df9b00    7           6a201c ?         0:00 syncer
  1 S           0   324     1  0 168 20        1019f0d00   24 400003ffffff0000 ?         0:00 inetd
1003 R          0    28     0  0 152 20        101b7a900    0                 - ?         0:00 vxfsd
  1 S           0   237     1  0 155 20        1019cb600   20           701ef0 ?         0:00 ptydaemon
  1 S           0   380   367  0 158 20        101b60500   48          32011c0 console   0:00 ksh
  1 S           0   250     1  0 127 20        1019f6d00   15           623a74 ?         0:00 nktl_daemon
  1 S           0   356     1  0 154 20        101e56800   19        101b76d2e ?         0:00 cron
  1 S           0   260     1  0 127 20        1019a5200   18           6f2e8c ?         0:00 ntl_reader
  1 S           0   261   260  0 127 20        1019f8b00   29        1019f75c0 ?         0:00 netfmt
  1 S           0   352     1  0 154 20        101e3d500   46           746ca4 ?         0:00 pwgrd
  1 S           0   359     1  0 154 20        101e5db00   14        1019a652e ?         0:00 envd
  1 S           0   400     1  0 154 20        1019a7f00   21           746ca4 ?         0:00 syslogd
  1 R           0   413   380  0 157 20        1019a7400   25                - console   0:00 ps
```

Not all flag bits are documented in ***ps(1)***; undocumented flag bits include:

040—Process' text locked in memory
0100—Process' data locked in memory
0200—Enables per-process syscall tracing
0400—Process has one or more lazy swap regions
01000—Process has 64-bit address space

This explains the 141 value seen for *init*: it has 0100 set because data is locked in memory, 040 because the text is locked in memory, and 1 because it's currently in core (0100 + 040 + 1 = 141). It also explains the 1003 value for system processes like *lvmkd* (01000 + 2 + 1), which in this example are 64-bit.

The list of non-system processes include:

- *init*
- *syncer*
- *inetd*
- *ptydaemon*
- *nktl_daemon*, *ntl_reader*, *netfmt*
- *cron*
- *pwgrd*
- *envd*
- *syslogd*

By examining the man pages available for these daemons we determine that we need most of them. As mentioned earlier, we can disable ***inetd*** if there are no ***inetd***-launched services. In theory, ***cron*** could be disabled if we do not plan to have any *cron* jobs, but this seems unlikely.

The ***envd*** process logs messages and can perform actions when over-temperature and chassis fan failure conditions are detected by the hardware. For example, in its default configuration it will execute ***/usr/sbin/reboot -qh*** when the temperature has exceeded the maximum operating limit of the hardware, in an attempt to preserve data integrity. We usually leave this daemon running, but we can disable its startup by modifying */etc/rc.config.d/envd*.

nettl is the network tracing and logging subsystem, and in the system default configuration starts three daemons: ***ntl_reader***, ***nktl_daemon*** and ***netfmt***. These are easily disabled by editing */etc/rc.config.d/nettl*; however we will lose potentially valuable log data, such as link down messages:

```
Apr  1 12:47:04 bastion vmunix: btlan: NOTE: MII Link Status Not OK - Check Cable
Connection to Hub/Switch at 1/12/0/0/4/0....
```

By default, console logging is enabled. Because there is little value in log messages being written to a console that is rarely looked at or may in fact be nonexistent, we can disable console logging. Disabling console logging causes the console filter formatter daemon, ***netfmt***, to not start:

```
# nettlconf -L -console 0
# nettl -stop
# nettl -start
Initializing Network Tracing and Logging...
Done.
```

The ***nettlconf*** command modifies the nettl configuration file, */etc/nettlgen.conf*, so this change will persist across system starts.

pwgrd is a password and group caching daemon. Since we have a very small password and group file it is unnecessary. Also, a little detective work with ***lsof*** and ***tusc*** (Trace UNIX System Calls)[9] shows us that it listens on a UNIX domain socket for client requests. We don't want to allow command channels like this to processes running as root, so we have additional incentive to disable it. To do so, we set the PWGR environment variable to 0 in */etc/rc.config.d/pwgr*:

```
PWGR=0
```

We also remove stale sockets, which will prevent unnecessary libc socket creation and requests to a nonexistent pwgrd listener:

```
# rm /var/spool/pwgr/* # really just need to remove status
# rm /var/spool/sockets/pwgr/*
```

ptydaemon is a mystery, since it does not have a man page. A little more detective work leads us to the belief that it may only be used by ***vtydaemon***, which we are not using. We decide to kill it and see if we can still login to the system remotely. (We temporarily enable ***telnetd*** to test this.) The remote login works fine, so we decide to permanently disable the startup of ***ptydaemon*** by setting the PTYDAEMON_START environment variable to 0 in */etc/rc.config.d/ptydaemon*:

```
PTYDAEMON_START=0
```

Next, we clean up the old logfile:

```
# rm /var/adm/ptydaemonlog
```

8. examine set-id programs

Many UNIX systems, including HP-UX, ship with numerous programs that are set-uid or set-gid. Some of these programs are not used or are only used by the root user, yet many of the vulnerabilities that are discovered in UNIX utilities rely on the set-uid root bit to raise privilege. We can improve the security of our system by removing these programs or by removing the set-id bit. To obtain a list of all files with either the set-uid or set-gid bit set on the system we can execute:

```
# find / \( -perm -4000 -o -perm -2000 \) -type f -exec ls -ld {} \;
```

We'll probably see well over 100 or so files listed. (In the sample configuration there are 145.) We may find two sets of LVM commands (in */sbin/* and */usr/sbin/*), each with more than 25 links that are set-uid root. Also, the SD commands are set-uid root. The following permission changes will greatly reduce the size of our set-id list:

```
# chmod u-s /usr/sbin/swinstall
# chmod u-s /usr/sbin/vgcreate
# chmod u-s /sbin/vgcreate
```

[9] tusc (Trace UNIX System Calls), syscall tracer for HP-UX, ftp://ftp.cup.hp.com/dist/networking/misc/tusc.shar

We may also notice there are some shared libraries that have the set-uid bit set. The reason for this is unknown, but it is safe to remove them. Interestingly, if we did not previously remove all saved patch files in */var/adm/sw/save/*, we see they have retained their set-id privilege. While this practice is questionable, these files are protected from being executable by non-root users because of the 500 permissions setting for the */var/adm/sw/save/* directory.

Our strategy is to remove the set-id bits from all files, then selectively add set-id back to just the few programs that need to be run by non-root users. For example, the following commands remove the set-uid and set-gid bits from all files, then add them back to *su* and the shared-library PAM version of the *passwd* command:

```
# find / -perm -4000 -type f -exec chmod u-s {} \;
# find / -perm -2000 -type f -exec chmod g-s {} \;
# chmod u+s /usr/bin/su
# chmod u+s /usr/bin/passwd
```

The file */usr/bin/passwd* has five hard links, and includes *chfn*, *chsh*, *nispasswd* and *yppasswd*.

The commands we choose to leave as set-id depend on the specific usage and policies of our bastion host. Let's say that the bastion host is a firewall gateway, where a few administrators will login via a unique, personal login, then ***su*** to root to manage the gateway. Here, */usr/bin/su* may be the only program on the system that needs to be set-uid.

In addition, a number of commands will function fine using default or commonly used options without privilege; some of these are ***bdf***, ***uptime*** and ***arp.*** However some functionality may be lost for non-root users—for example, we can no longer specify a filesystem argument for ***bdf***:

```
$ bdf /dev/vg00/lvol3
bdf: /dev/vg00/lvol3: Permission denied
```

9. examine file permissions

A freshly installed HP-UX system will contain a number of files that are writable by other. (That is, the 002 bit is set in the mode bits.) These files can be listed with the following command:

```
# find / -perm -002 ! -type l -exec ls -ld {} \;
```

We don't display symbolic links with the write other bit set because the mode bits are not used for permission checking.

One approach is to remove the write other bit from all files, then selectively add it back to those files and directories where it is necessary. The following command can be executed to remove the write other bit from all files that have it set:

```
# find / -perm -002 ! -type l -exec chmod o-w {} \;
```

Now we open up the permissions of files that need to be writable by other users:

```
# chmod 1777 /tmp /var/tmp /var/preserve
# chmod 666 /dev/null
```

Note that we also set the sticky bit (01000) in publicly writable directories like */tmp* and */usr/tmp*. This prevents unprivileged users from removing or renaming files in the directory that are not owned by them. (See ***chmod(2)***).

10. perform security network tuning

HP-UX 11 introduces the ***ndd*** command to perform network tuning. ***ndd -h*** produces a list of help text for each supported and unsupported ***ndd***-tunable parameter that can be changed. After examining this list, we decide the following are candidates for changing on a bastion host:

Network device	**Parameter**	**Default value**	**Suggested value**	**Comment**
/dev/ip	ip_forward_directed_broadcasts	1	0	Don't forward directed broadcasts
/dev/ip	ip_forward_src_routed	1	0	Don't forward packets with source route options
/dev/ip	ip_forwarding	2	0	Disable IP forwarding
/dev/ip	ip_ire_gw_probe	1	0	Disable dead gateway detection (Currently no *ndd* help text; echo-requests interact badly with firewalls)
/dev/ip	ip_pmtu_strategy	2	1	Don't use echo-request PMTU strategy (Can be used for amplification attacks and we don't want to send echo-requests anyway)
/dev/ip	ip_send_redirects	1	0	Don't send ICMP redirect messages (if we have no need to send redirects)
/dev/ip	ip_send_source_quench	1	0	Don't send ICMP source quench messages (deprecated)
/dev/tcp	tcp_conn_request_max	20	500	Increase TCP listen queue maximum (performance)
/dev/tcp	tcp_syn_rcvd_max	500	500	HP SYN flood defense
/dev/ip	ip_check_subnet_addr	1	0	Permit 0 in local network part (Should be the default)
/dev/ip	ip_respond_to_address_mask_ broadcast	0	0	Don't respond to ICMP address mask request broadcasts
/dev/ip	ip_respond_to_echo_broadcast	1	0	Don't respond to ICMP echo request broadcasts
/dev/ip	ip_respond_to_timestamp_ broadcast	0	0	Don't respond to ICMP timestamp request broadcasts
/dev/ip	ip_respond_to_timestamp	0	0	Don't respond to ICMP timestamp requests
/dev/tcp	Tcp_text_in_resets	1	0	Don't send text messages in TCP RST segments (should be the default)

Some of the default values match our preferred value, but we can choose to set them anyway, just in case the default should change in a future release. ***ndd*** supports the ***-c*** option, which reads a list of tunables and values from the file */etc/rc.config.d/nddconf*, and which is run automatically at boot time.

However, there are some problems with the default setup. First, at the time of this writing, ***ndd -c*** is able to handle only 10 tunables in *nddconf*. Moreover, ***ndd -c*** is run at the end of the net script, which is after network interfaces have been configured. One problem with this is that it is too late to set *ip_check_subnet_addr* if we are using subnet zero in the local part of a network. But more importantly, we want to set tunables before the network interfaces are configured.

Note: The ordering problem has been fixed in a recent transport patch, but the limit of 10 tunables remains.

We can use this workaround, with a new startup script and configuration file:

```
# cp /tmp/secconf /etc/rc.config.d
# chmod 444 /etc/rc.config.d/secconf
# cp /tmp/sectune /sbin/init.d
# chmod 555 /sbin/init.d/sectune
# ln -s /sbin/init.d/sectune /sbin/rc2.d/S009sectune
```

We run the script immediately after ***net.init***, which sets up the plumbing for the IP stack, then runs ***ndd -a***, which sets transport stack tunable parameters to their default values. (The ***sectune*** command and a sample ***secconf*** are available for download.[10])

11. install software and test the configuration

At this point we can install, test and configure the application software that we will use on the bastion host, such as the BIND product, a web server, a firewall product, etc. Security software, such as SSH (Secure Shell) and TCP wrappers can be installed at this point, as determined by the specific security requirements and use of the bastion host. Again, we need to exercise extreme caution when installing new software on our bastion host. We try to always get the latest version of the product, one that has been patched against all known security defects. We may even install the product first on another system and determine if it can be secured. We try to think like an attacker, and ensure that the bastion host is able to protect itself with the product installed.

12. create a system recovery tape

Finally we create a bootable System Recovery Tape of the root volume group; this tape can also be used to clone the system to other hardware that is supported with the same software configuration (for example, we can clone from an HP L2000 to an N4000).

The following can be executed online (very cool), though we will want the system in a somewhat quiescent state:

```
# /opt/ignite/bin/make_recovery -Ai
      Option -A specified. Entire Core Volume Group/disk will be backed up.

      ************************************
       HP-UX System Recovery
       Going to create the tape.
       System Recovery Tape successfully created.
```

[10] Sample secconf and sectune scripts, http://people.hp.se/stevesk/secconf and http://people.hp.se/stevesk/sectune.

conclusion

With the simple methodology presented, a paranoid mindset, a little detective work and some persistence, it's relatively straightforward to construct a robust bastion host using HP-UX.

references

- Marcus J. Ranum, "Thinking About Firewalls," SANS 1993. An updated version, "Thinking About Firewalls V2.0: Beyond Perimeter Security," is available at http://pubweb.nfr.net/~mjr/pubs/think/index.htm.
- Elizabeth D. Zwicky, Simon Cooper, and D. Brent Chapman "Building Internet Firewalls, 2nd Edition," O'Reilly & Associates, June 2000.
- HP Security Bulletins are available at http://us-support.external.hp.com/ and http://europe-support.external.hp.com/. Select "Search Technical Knowledge Base." You need a login to access security bulletins, but you can register for one in a few minutes.

for more information

Looking for more information about HP-UX 11 and HP-UX 11i? Visit these Web sites:

- http://www.docs.hp.com
- http://www.hp.com/go/hpux

For more information, contact any of our worldwide sales offices or HP Channel.

For the location of the nearest sales office call:
United States of America: +1 800 637 7740
Canada:
Hewlett-Packard Ltd.
5150 Spectrum Way
Mississauga, Ontario L4W 5G1
+1 905 206 4725
Japan:
Hewlett-Packard Japan, Ltd.
Japan Country H.Q.
3-29-21, Takaido-Higashi, Suginami-ku,
Tokyo, 160-8585 Japan
+81 3 3331 6111
Latin America:
Hewlett-Packard
Latin American Region Headquarters
Waterford Building, 9th Floor
5200 Blue Lagoon Drive
Miami, Florida 33126 USA
+1 305 267 4220
Refer to country phone numbers
Australia/New Zealand:
Hewlett-Packard Australia Ltd.
31-41 Joseph Street
Blackburn, Victoria 3130
Australia (A.C.N. 004 394 763)
+61 3 9272 2895
Asia Pacific:
Hewlett-Packard Asia Pacific Ltd.
17-21/F, Shell Tower
Times Square
1 Matheson Street
Causeway Bay
Hong Kong
+8522 599 7777
Europe/Africa/Middle East:
Hewlett-Packard S.A.
150, Route du Nant-d'Avril
CH-1217 Meyrin 2
Geneva, Switzerland
+41 22 780 81 11
European Multicountry: +41 22 780 81 11
Middle East and Africa: +41 22 780 71 11
European Headquarters: +41 22 780 81 81
Refer to country phone numbers

For direct country contact call:

Argentina: +541 787 7145

Austria: +43 1 25 000 0

Belgium and Luxembourg: +32 2 778 31 11

Brazil: +5511 7296 8000

Chile: +562 203 3233

Colombia: +571 629 5030

Denmark: +45 45 99 10 00

East Central Europe, CIS, and Yugoslavia: +43 1 25 000 0

Finland: +358 9 887 21

France: +33 1 69 82 60 60

Germany: +49 7031 140

Greece: +30 1 689 644

Hungary: +36 1 252 7300

Iceland: High Performance Systems hf. +354 1 67 10 00

Ireland: +353 1 615 8200

Israel: Computation and Measurement Systems (CMS) Ltd. +972 3 5380 333

Italy: +39 2 92122770

Mexico: +525 326 4600

Netherlands: +31 20 547 6911

Norway: +47 22 7356 00

Poland: +48 22 608 77 00

Portugal: +351 1301 7343

Russia and the CIS, excl. Ukraine: +7 095 923 5001

Slovenia: +38 61 55 84 72

Spain: +34 1 631 1600

Sweden: +46 8 444 2000

Switzerland: +411 735 7111

South Africa: Hewlett-Packard South Africa (Pty) Ltd.+27 11 806 1000

Turkey: +90 212 224 5925

United Kingdom: +44 1344 369231

Venezuela: +582 239 4133

Full information on HP products is available at **www.hp.com**

M0800

invent