



Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permitted without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Security Essentials: Network, Endpoint, and Cloud (Security 401)"
at <http://www.giac.org/registration/gsec>

Overview

Much documentation has been written about security including its policies, procedures, forensics and the strategies for protecting technology from unauthorized access, yet little is seen, outside of the vendor forums about the Hewlett Packard family of Unix servers. To this end, the following description, for the layman and the technical person both, a brief outline of the steps used in securing an HP server and some to keep in mind when dealing with security on different unix flavors.

The parameters

We propose to secure a server that will house a corporate database with storage for 300 employees and 52 weeks of forecast data, it will include over 200 MB of disk space, also configured will be a dual-port GSC 802.3 LAN card and RS-232C console and UPS ports. A midrange server for this size database will make use of 4-120Mhz PA-RISC processors, standard in the HP-UX family of servers, 4 Gigs of memory; the node will be configured within as LAN environment behind redundant firewalls with 24x7x365 availability and a SLA requires 99.999% uptime. Generally, an Oracle DB environment can be embedded in a three-tier environment with an NT front end, and an HP-UX Unix server that serves as the application server.

The Process

We will begin building our secure server by installing the Operating System from scratch via vendor supplied CD's and be selective with the number and types of system utilities that get initially installed; many defaults configurations will be accepted and opt to disable or remove unwanted apps later. We will then apply, from the most current distribution, the recommended hardware, software, security and general release patch bundles specific to our configuration.

Once our server is built, our first security measure will be to implement access authentication by way of the 'Trusted Systems' utility, an HP-UX OS-included security application that provides password policies and auditing; next we will be removing unneeded pseudo accounts and users and increase the sensitivity to file user and group permissions system and world files. We will investigate SAM, for 'kernel tuning', spend some time on OS patching, network services, and sendmail; brief notes on backup strategies and disaster recovery will also be provided. Finally, we will make some comments on backup and disaster recovery strategies and conclude with a comment on 'canned' security applications available on the internet. A good source of information for this process is the HP-UX 'instant documentation CD' provided freely with the distributed OS, in it you will find user guides, manuals and other technical documentation for the administration of these unix family of servers.

Installing the OS

The process of installing the OS from scratch should take place with the server disconnected from the network. We will be making use of an HP9000-K420 4-way CPU with 1152 Mb RAM, 12 Gigs disk space (2-4gig and 2-2gig disks are available for this) and 2-100 Mb GSC NIC's and a DDS-2 tape drive, server. We have selected Nov. 1999 Core IO OS and plan to install it as a 32bit OS, the 64bit option can alternatively be selected if you are interested in installing the web enabled version of Oracle. We plan to update the OS by way of patches and application enhancements.

Building the OS is accomplished through interactive screens, at every step of which a different parameter is set, the following steps are taken:

1. Boot from selected Core-IO CD after powering-up the server
2. On first screen select the OS you wish to install, we select: HP-UX B.11.00 Default

Note: We are not selecting the advanced method for loading the OS, this will cause additional applications to be installed by default. This will also simulate a real-world situation where applications, utilities and processes are preloaded and have to be removed.

3. Select 32-Bit HP-UX without CDE support, for the environment.
4. Next select default root disk; we will make use of the 1st 4GB available disk.
5. Root swap area is selected next; our value will be the default 1GB. Value can be changed later as needed.
6. The file system selected is, Logical Volume Manager (LVM) with VxFS. Most logical volumes housing the operating system will have this format and be located in the root volume group vg00 will be created using this format.
7. The language selected is English
8. A 2-User default license will be selected next.
9. Additional software selections can determine hardware/software requirements later, be selective in adding these.

The server will reboot upon accepting the above selections. The next step will be to initiate configuration of the OS by optioning hostname, timezone, date_time, root_passwd, ip_address, addl_network, these system parameters require planning and should be compiled prior to the server build in coordination with management and while adhering to corporate policy and naming convention. The boot process, lasting between 15 and 30 minutes, loads the OS, scans hardware building a device file list and initiates start-up scripts according to configured requirements, the above process can also be referred to as: building or loading the system kernel. The system is finally available when we see the prompt: Console Login:, continue by login on as root with the password you provided above.

Securing the server

Password security is the first and most powerful line of defense, because of this we implement 'trusted systems'. By making use of the: `/usr/sbin/tsconvert` command, we are creating a protected database on the system for storing security information. This process will literally convert the system to a secure server, shadowing the passwd file and implementing password aging; auditing processes can and should be also enabled at this time by selecting different options available, again, consult corporate policy. Password behaviors and file locations will be changed and defined according to your selections, for instance, the trusted audit log will be located at: `/.security/etc/auditfile1`. Be careful, all UID's other than root will be forced to change their passwords the first time they log on, so be prepared to provide procedures and documentation instructing users. The installed umask of 0 for all accounts will be changed to 07077, and the password section of `/etc/passwd` will be replaced by '*' masking the encrypted string; a word to the wise, use the `sam` utility and grant someone shutdown rights in case the root account becomes deactivated, in HP-UX when a server is rebooted, the administrator can bring up the server in 'single-user mode', this mode allows the root user the availability to login as root without entering its password and changing it. The secure database will be located in `/tcb/files/auth/system/default`, and depending on your version of OS, the passwd files may be located in `/tcb/files/auth/[a-z, A-Z]/*`; each user in the original passwd file, will have a separate directory.

Once the server is secured in this fashion, the next step we'll take is to remove any unneeded pseudo-accounts created by the install process so we take a look at system groups and users that do not start any particular processes, we do this by browsing the password and group files searching questionable entries, after a little verification we target groups: lp, uuucp and daemon. Try the following commands:

```
find / -group lp -o -group uuucp -o -group daemon -o -group -exec ls -ld {} \;
```

The output from the above command could be extensive so be prepared, if you find groups that do not start any processes, then remove them by:

```
groupdel lp; groupdel uuucp; groupdel daemon;
```

The same process can be used for user accounts:

```
find / -user uucp -o -user lp -o -user nuucp -o -user www -o -user daemon -exec la -ld {} \;
```

Again, this output could also be somewhat extensive, extraneous users can be removed by:

```
userdel uucp; userdel lp; userdel nuucp; userdel hpdb; userdel www; userdel daemon;
```

The remaining user default accounts (bin, sys and adm) should be configured with an invalid login shell path like /; and changing their shell to using a *noshell* program.

Now locate all setuid programs on the system by the following command:

```
find / -type f -perm -4000 -exec ls -l {} \;
```

; then make a decision as to whether they really need this much control, you may have some flexibility on the above, some setuid *root* programs may be made setuid to some other user or groupid created especially for the purpose.

The topic of file and directory permissions can be summed best by a recent comment donated into a technical forum '...it is easier to 'open up' permissions later than to have to tighten them up after the fact...'. In this case the technite suggests setting up umask permissions when first building the server rather than later. Root's umask should be verified to be 022, but most preferable would be 077; regular users umask should be set, as a minimum, to 022 in the /etc/profile so that all users have protection for their own files. A fatal mistake in securing this server would be to have roots' umask set to 666 or 777, this would imply that anyone, not just root, can modify or remove system files, or possibly worse, run a root script or program compromising the environment. The umask for csh and tcsh shell users can be set in the /etc/login file. Use the following lines in the users profile to set the umask:

```
export WHO=`whoami`; if [ "$WHO" = "root" ]; then; umask 077; else; umask 027; fi;
```

remember each member also has a profile in the \$HOME directory and can alter their own umask by executing the command. Alternatively, you set the umask for root and daemons by putting a umask 022 command in /etc/rc after the: PATH= and HOME= lines. In the /etc/rc.local file change the permission setting for /etc/motd by changing the line: chmod 666 /etc/motd to chmod 644 /etc/motd. Like the system banner, /etc/motd can be abused by unscrupulous people.

SAM; the System Administration Manger is an HP-UX tool that provides an easy-to-use user interface for performing setup and other essential system tasks. This interface is the alternative to the command line interface that has always been available for all flavors of Unix. A system administrator can provide restricted superuser access to other administrators by configuring the interface by typing: *sam -r*. SAM also provides a "Tuned Parameter Set" for most standard system applications, yet, as these parameter sets were made for generic system installations they should be used with caution. Kernel changes are made by entering SAM, typing *sam* and selecting "Kemel Configuration" from the main menu and the "Configurable Parameters" option; of the approximately 110 'tunable' parameters, only a few are critical to any installed device or application, also, be aware that after every change in this area the system will request to be rebooted for all configuration changes to take effect.

Patching the Server

Patching is a critical part of the HP-UX platform, generally the vendor suggests a quarterly analysis be performed for all sorts of hardware compatibilities, software enhancements and security updates. We used the September 2000 HP-UX 11 diagnostics and tested patch bundles CD to install the Software General Release patches: XSWGRI100 and the Software and Hardware Critical Patch bundles: XSWHWCRI100 as described. Note: the *commit_patches* option will be used later, do not opt for it now, committing patches is a process that the admin uses for cleaning up and availing necessary disk space. Patches installed on our sever should be analyzed later, some of these may not be necessary and reading the *patch.text*, provided with each patch for details about the patch dependencies can lead to less headaches. It is OK to leave unknown installed patches in place or to install a particular ones that you are not sure about, the space the majority of them take is minimal and they will not affect the performance of server. To install patches on the server,

perform a scan of available devices by way of the `ioscan -funC disk` command. Mount the CD on the `/cdrom` or `/SD_CDRROM` (software distributor recommended mountpoint) by way of the `mount /dev/dsk/c3t2d0 /cdrom`, and run the `swinstall` application and follow directions.

Security patches, on the other hand can be verified on the platform by making use of the “`security_patch_check`” utility available from HP at: http://www.software.hp.com/cgi-bin/swdepot_parser.cgi/cgi/displayProductInfo.pl?productNumber=B6834AA. It is a Perl script that analyzes file sets and patches on an HP-UX system and it is a tremendous tool that helps HP-UX administrators ensure their systems' security patches are up-to-date; released on May 9th, 2001 by HP, please install and use.

Remove `/etc/hosts.equiv` and `/.rhosts` files unless you have some overriding need for them, if so, remove all entries for root in them and use them in the individual users home directories. Set `/.rhosts` so that no other machine is equivalent and test the functionality for the ability to use: “* -user name” to deny trusted logins from any system and username pair. Verify that hostnames listed in the `/.rhosts` files have fully qualified names firsts and then aliases, they should contain NO ‘+’ entries. Do not forget to check the `.netrc` file as it has similar effects to FTP access, whenever possible, remove it.

Do not allow direct `root` logins, except maybe from the `console`, this is performed by adding the word `console` into the `/etc/securetty` file, other terminals must `su` to the superuser account upon login. Limit the users who are allowed to `su` to `root`. By inserting: ‘`auth.notice /var/log/authlog`’ into the `/etc/syslog.conf` file, the admin can monitor `su` activity, have `sendmail` send you the `su` log from time-to-time. The admin can implement `sudo` in place of `su` to avoid giving people unrestricted root access.

The stickybit is another one of those extremely useful security utilities. If is set on a directory, users will not be permitted to delete files, which they do not own. The importance: utilities create temporary files in a public directory and assume that what they read back will be what they wrote. A malicious user can replace the temporary file with questionable code that can execute inadvertently: `chmod +t <directory name>` will work just fine.

The internet services daemon, `inetd`, controls most –but not all-- of the services your system provides to the rest of the world. Edit the `/etc/inetd.conf` and disable (by placing a “#” in column 1 of each line) all services which you do not plan to use. We should also enable `inetd` logging as `inetd` will remain enabled; add the `-l` argument to the `INETD_ARGS` environment variable in the `/etc/rc.config.d/netdaemons` as follows: `export INETD_ARGS=-l`. Lets now look at each one of these services available in a default configuration:

ftp - If you don't plan to use FTP, disable it, if you do, consider using an FTP daemon that has added logging and access control features such as `proftpd`.

telnet, shell, login, exec – These utilities allow remote users to log into any one of your systems, chances are you need some of these, consider using `ssh`.

talk – Cute, use the phone, disable it.

uucp – If you don't use it, disable it.

tftp – The Trivial File Transfer Protocol, there is no place for this service in a production server, disable it.

finger – This service is part of many root tool kits. Disable it, very dangerous.

netstat – This service is necessary to troubleshoot network issues, if possible use the TCP Wrapper on it.

ntpd – Sets local date/time by polling Network Time Protocol server. Use the start-up script method instead of the `cron`.

echo, discard, daytime, chargen – Depending on who you speak with, these are dangerous, used in SYN-ACK DoD attacks, personally I would disable them, this would cause developers from charging my office.

rex – Scary, rpc based remote execution server, should we say more?

walld – Although useful in certain situations, best safe with TCP Wrapper. See talk above.

tooltalk – Used by many common desktop elements, yet serious remote exploits have been created targeting this utility, can be disabled.

uucp – a suite of utilities, which allow a primitive form of networking using hardwired RS232 cables and dialup telephone connections. If not used, disable it by: editing the `/etc/passwd` file and setting the encrypted password field to “*” for the `uucp` account, the shell field should also be edited to say: `noshell`.

Note: It may be a good idea to turn off as many execute permission on related commands as listed above. This same approach can be taken with other standalone daemons not started by `inetd` and where TCP Wrappers have no effect. Execute `ps -ef`, review the display and kill any processes that ask for it and review the fallout, act accordingly, one suggested activity may be to rename, unlink or remove permissions of their start-up scripts.

A word about TCP Wrapper; written by Wietse Venema and used as a tool commonly used on Unix systems, it is configured to monitor and filter connections to network services. It is the preferred method for restricting network services, it permits the admin to specify access control lists by site, domain or usernames while logging all requests. The most (and safest version) of the utility can be obtained at:

<ftp://ftp.porcupine.org/pub/security/>, it is encouraged that any download after Thursday, January 21, 1999 at 06:16:00 GMT of this utility be verified for its authenticity as it is known to have been compromised. Best option is to obtain information from the above site and CERT advisories and proceed accordingly.

The `nsswitch.conf` DNS resolver can be set to search local files only, if possible; also `chmod 444 /etc/nsswitch.conf` and the `/etc/resolv.conf` files for their protection.

NFS is a notorious security problem, if it is going to be used, use the `nosuid` and `readonly` options.

Furthermore, make use of specific entries in the `/etc/exports` by way of: `-access=hosta:hostb`.

Disabling NFS and NIS from starting up at boot time can easily be accomplished by vi'ing the `/etc/rc.config.d/nfsconf` and the `/etc/rc.config.d/namesvrs` files and setting `NFS_CLIENT=0` and `NFS_SERVER=0` and `NIS_MASTER_SERVER=0`, `NIS_SLAVE_SERVER=0` and `NIS_CLIENT=0`, respectively. These settings can be set manually =1 if done through SAM for the purposes of temporarily mounting or exporting filesystems. Remember to disable them back when done with the operation. Do not selfreference an NFS server in its own exports file either by name or by the loopback address: `localhost`.

Network card drivers are next; these are also available on the app cd's and should be installed at this time. Remember that it must be configured according to the switch or router's port configuration, suggested is 100Mbps running at Full-Duplex with `auto-negotiation=OFF`. Most of these changes can be done with the 'lanadmin' command-line utility, see the man pages for lanadmin for more information.

No document on unix security would be complete without a quick study of Sendmail. In searching one of my systems I verified the latest release of sendmail, v8.9.3; the latest available (as of this writing), v8.11.3 can be found at: <ftp://ftp.sendmail.org/pub/sendmail>, it comes complete with release notes. To verify your version, type: `telnet <local host> 25`, this will connect you to the SMTP port 25 of your system. The following should return: `220-<fully qualified name> Sendmail 8.9.3 Patch () 8.8.6 Ready Wed May 8, 2001`. A superficial test to verify your susceptibility to compromise can be tested by running the command: `wiz`, the response should be: `500 Command Unrecognized`; next type: `expn`, the response should be: `502 Sorry, we do not allow this operation`. If the response is: `501 Argument Required`, then rest assured that your server could be compromised. Next type: `quit`, to exit.

Backup and Disaster Recovery Strategy

As we continue our secure server study, we describe data recovery from the standpoint of a contingency plan and in this scenario, backups play the most critical role while cost of time and resources play a back role. Realistically, though, few organizations are fully prepared for all possible events that can arise from disaster, denial of service or something as simple as peripheral failure.

We could repeat details on system utilities like tar, dump, cpio and dd, or describe the usage of applications for different platforms, namely, ArcServer for NT or RBMS for IBM, but are more inclined to provide information on a solution that we find extremely effective for this platform, i.e.: HP OpenView' enterprise backup software, Omniback II. Although the application is installed from standard release Application CD's, its license is controlled by use of 'code words' provided upon purchase; a 30-day evaluation-emergency license can be installed with the aid of standard technical support.

OpenViews' *Omniback II* is a robust server based application that can be considered secure, few advisories have emerged that detail incidents involving the compromise of either its server or client based modules. Its internal database manages datalists, schedules, tape management rules and so on; it also has much room for scaling and cross platform implementation. While the client installation makes use of server-to-server root authentication, it depends of an administration GUI that verifies fully-qualified node names. Day-to-day activities are managed from configuration files, via the GUI, that are securely embedded in deep directory structures.

DLT, the media of choice (although DDS devices can also be configured), cartridges are monitored for expiration, state and capacity within the application. Tape management is performed by accessing media libraries (20-50-100 tape cabinets) via a GUI based environment and is configurable for son-father-grandfather type media rotation. Most installations include tape, 3 to 6, drive cabinets that make use of SCSI technology; fibre options can be purchased for increased throughput to the media. A most reliable configuration makes use of parallel writes onto two different tape drives, in this method, one tape is maintained on site for quick on-the-fly type recoveries while another copy is sent off-site on a daily basis. This, in addition to Omniback's ability to write individual tapes from six different servers, synchronously, makes the application full proof as the enterprise solution of choice for contingency recovery. As mentioned earlier, the great majority of media used by OmniBack is maintained on-site within the DLT Tape Exchanger (70%), while the remaining tapes are packaged and set away for off-site storage. Off-site storage is part of the disaster recovery process that safeguards data and documentation from damage in case of accident, disaster or vandalism.

A complete HP-UX disaster recovery strategy, from the point of HP-UX system tools and applications, can be said to be two-fold: Omniback II for backup purposes and a system recovery tool available for the platform, *Ignite-UX*. The system recovery feature of Ignite-UX implements an easy-to-use, consistent and reliable process for recovering the root system disk or volume group on HP-UX systems by way of the "make_tape_recovery" utility. The System Recovery process allows a user to cold install a failed system from a "recovery tape"; currently DDS while a DLT method is currently in beta. With advance planning, it recreates the "Core OS" portion of the root disk or root volume group from the recovery tape, and gets the system back to a fully customized system without the user having to go through all the processes or reconfiguring the system, as in the case of a standard cold install with the HP-UX distribution media, adding patches, subsystems, selecting software, etc. The make_tape_recovery utility can be scheduled by way of root or an admins' cron.

All or most methods discussed can be considered critical in safeguarding communications to and from a production server. Any host connected to a network is vulnerable to compromise by individuals or code

introduced by an automatic method or personal access, malicious code can be introduced through HTTP Java applets, Trojan executables and SMTP attached files.

Conclusion

This document is intended for system administrators, users and managers; to increase the sensitivity about their responsibilities in making a computing environment secure from malicious attacks from outside as well as inside an organization. Secure environments can be accomplished by the use of many 'canned' applications and utilities readily available on the internet, two of the best: **Crack**, a password cracking utility that checks for poor passwords and **Tiger**, a utility that can help in checking for variances and possible holes in: cron entries, mail aliases, NFS exports, inetd entries, .rhosts & netrc files, provides for one of the best ways of performing an internal audit. These scripts expand on and are similar to the Computer Oracle and Protection System (**COPS**) scripts which is a security status checker, essentially, it checks various files and software configurations to see if they have been compromised, (as would happen when they are edited to plant a trojan horse or back door), and checks to see that files have the appropriate modes and permissions set to maintain the integrity of your security level. Additionally, Secure Shell (**ssh**), a utility used to connect, login and execute commands in a remote system and which provides strong authentication and secure communication over unsecured channels.

The art of securing the unix servers is without a doubt an extensive proposition, but while it is fun, it is time consuming and requires support from administration teams, management and much technical documentation. It is hoped that with the help of the presented process, the use of the readily available utilities and resources we have provided direction in the method for securing an HP-UX server.

References

- [1] Kevin Steves, "Building a Bastion Host Using HP-UX 11" April 2000
stevesk@sweden.hp.com, Hewlett-Packard Consulting, Sweden
- [2] Julie Geer-Brown, "Oracle Ships Industry's First Complete Data Protection Solution"
<http://www.oracle.com/corporate/press/index.html?622119.html> March 17, 2001, Oracle Corporation
- [3] CERT[®] Advisory CA-1999-01 Trojan Horse version of TCP Wrappers, January 22, 1999,
<http://www.cert.org/advisories/CA-1999-01.html>
- [4] 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/.
- [5] HP-UX Patch Security Matrix, ftp://europe-ffs.external.hp.com/export/patches/hp-ux_patch_matrix.
- [6] Computer Incident Advisory Center, System Monitoring Tools-
<http://ciac.llnl.gov/ciac/ToolsUnixSysMon.html>
- [7] HP Online Technical Documentation is located at: <http://docs.hp.com>.

ⁱ <http://forums.itrc.com/cm/QuestionAnswer>