Global Information Assurance Certification Paper

Analysis of Primary Web Servers at

GIAC Enterprises

November 22, 2000

Matt Miller

SANS Practicum, Monterey 2000 conference
GIAC SECURING UNIX PRACTICAL ASSIGNMENT

Executive Summary of primary UNIX web servers

The two primary Unix web servers are configured identically which helps simplify administration. Steps have been taken to enable secure access to these servers; however, secure access is not required. Standards for the installation of a Unix server are stated, but it is easily possible to miss a step. If the suggested steps in this document are taken, the initial installation of the servers will be even more complicated, with the chance of human error being greater. We suggest compiling these changes into a JumpStart installation and/or post installation scripts for ease of administration and to enable a quicker recovery from a total system failure.

Some of the most serious security holes are the easiest to fix and these are prioritized on the next page. Basically, you need protection against password "sniffing" of user and root accounts as well as data encryption between the clients and the server. The normal web server data can go out unencrypted because it is meant to be publicly accessed data. The big concern on these servers is the administrative connections made to the machines from the office, or from a remote location.

Securing and encrypting access to the servers will only help if your passwords are good. If they can be guessed, then all other measures are null and void. Administrative passwords should be unique, and all actions should be logged as much as possible. Logfiles must be checked regularly for potential problems and automatically reported to the proper people.

Many attacks are aimed at security holes in server services (daemon processes.) To protect against these attacks, keep your operating system and applications patched to current revisions, and don't run any services you don't need. This sounds simplistic, but the less services you have running, the less you have to worry about patching the server.

Once these "perimeter" steps are taken, the rest of the steps are just there to make it more difficult for the potential hacker to utilize the system without being discovered. Hopefully this will also delay the hacker long enough for you to take action to lock them out.

Once an attack is detected, you must have procedures in place to stop further damage. Without prior knowledge of the needed utilities, you will not be able to act fast enough to counteract their attack. (Once rootkits are installed, they can move very quickly!)

Any future applications added to these servers should be audited for new ports that are opened. The log files from any new services should be sent to the syslog server suggested below. Also, if any application servers or database servers are added to support the primary web site, measures should be taken to encrypt the data that goes between them and these primary servers.

Operating system vulnerabilities

The choice to use Sun Solaris servers (on SPARC machines) was made due to the fact that this platform has been in use, and well tested, for many years. The long history of this platform suggests that many tunings and patches have been developed to increase security and performance over the years; however, it should be noted that this prolonged access to the software has given hackers many years of experience as well. This platform does have an advantage over its counterparts that run on Intel processors; the hardware is more expensive, and this requires a larger investment from potential hackers. (Solaris is now available for Intel processors, so this advantage is diminishing.)

Patches are applied monthly (from http://sunsolve.sun.com/) to the Solaris servers to insure the latest in performance and security. http://www.securityfocus.com/ is also checked at least weekly to stay on top of urgent security issues that need to be patched.

Configuration vulnerabilities

Current practices for building Solaris servers include installation of the following:

- The Developer package of Solaris off of the OS CD
- Local authentication only (no NIS)
- Sshd (with sftpd) from ftp://ftp.ssh.com/pub/ssh/
- top, and gzip packages from http://www.sunfreeware.com/
- Any patches available from http://sunsolve.sun.com/
- Root login disabled (must su for root privileges)
- Core files are enabled
- 3rd party software listed in the next section

All logging is done with default measures on the local machine. Password expirations are not set; it is up to the administrator to remember to change them. There are two groups of people who know the root password (the applications group and the server support group.)

The current security risks below are listed in order; the easiest to fix and most crucial items are listed first:

- Telnet/FTP are still enabled, even though secure equivalents exist
- Root passwords are not unique, and many administrators have access to the root password.
- No regular checking of logfiles is done; no automated logfile utilities in use.
- Sendmail is running in the default configuration without smrsh.
- Many unnecessary services are running which give out system information to remote users. (This information can help stage a more appropriate attack.)
- Administrative web sites not requiring SSL connections.

- No sudo or tcp wrappers are used (limited logging capability)
- Core files enabled (shadow file contents are in these files)
- No audit software is in use (like Tripwire or Tiger)
- No secure remote logging server for log comparisons
- Backup tapes are in the data center with the servers; a fire could destroy all copies of the data. Also, the tapes should be moved to a locked cabinet or vault in a new location.
- Compilers installed on both systems

Suggested changes/additions to procedures:

- Remove telnet, ftp, finger, ruserd, rquotad, rstatd, walld, printer, talk, login, and echo lines from the /etc/inetd.conf file
- Rename (mv * to disabled_*) the links in /etc/rc3.d for the following services: nfs.server, snmpdx
- Rename (mv * to disabled_*) the links in /etc/rc2.d for the following services: preserve, sendmail. Preserve is rarely needed and can be used to gain access to a root shell by an unprivileged user. The only thing you loose is the ability to recover data from an editor session when a system crashes. Sendmail is a security risk that can be easily secured by only running it if you need it, and running it in a special shell called smrsh. (This is the closest equivalent for using chroot for other processes because sendmail must run as root.) From cron, you will need to run sendmail with the "-q" option to flush the queues.
- Consider removing X server packages if all administration is done remotely (only leave X client packages) The X server is just another service with a security risk and it also consumes a lot of resources. If you don't use graphical applications locally, remove it.
- Install the sudo package and give administrators access to only what is needed. Limiting users through sudo (even administrators) to only what they need access to helps in two ways. It keeps a log of their actions and it reduces the impact if one of their logins is compromised. (Be sure to pay attention to utilities that can spawn a root shell!)
- Implement a secure logging server so that logs can be systematically compared for security breaches. You will need to run syslog-ng from http://www.balabit.hu/products/syslog-ng/ to be able to send messages over TCP, which should then be tunneled through SSH. NTP clients should also be loaded on all servers to make sure the time stamps in the logs can be compared accurately.
- Disable Core file dumps by adding "set sys: coredumpsize=0" in /etc/system
- Re-compile (if necessary) SSH to include tcp wrappers and set PermitRootLogin to NO in the /etc/sshd_conf file. Both of these options will allow for better logging.

The changes suggested above follow the simple guidelines of removing what you don't need, and only giving access to what is needed. All services have a security risk

associated with them, so eliminating services that are not needed lowers your overall security risk.

Risks from installed third-party software

Software in use on the 2 servers in this audit:

- iPlanet Web server (load balanced through IP switch) running as webuser
- iPlanet Directory Server (load balanced through IP switch) running as diruser
- Omniback backup client

Suggested changes/additions to procedures:

- Authentication logs from the iPlanet servers should be sent back to the suggested syslog server.
- Administrative web sites should be limited to SSL connections only to protect passwords from plain text transmissions.
- Omniback backup client should be configured to send traffic over an SSH forwarded port to encrypt the data. Tapes should be stored in a secured area because they contain backups of /etc/passwd and /etc/shadow.
- LDAP communications should be tunneled through a SSH connection for encryption between the web server and the LDAP database.

Administrative practices

Current practices:

- These servers are updated at the same time, with the exact same patches to insure the configurations are the same.
- Root passwords are the same on both servers.
- Some administrative scripts have been created with setuid to root.

Suggested changes/additions to procedures:

- Install tripwire to monitor your system executables for possible Trojan horse attacks. This should be done after the other installs, just before connecting to the network. ("tripwire –m i" will initialize the database so you can check against it on the next run.)
- Run tara (from http://www.securityfocus.com) and take appropriate actions. See the section below on "Other issues/vulnerabilities."
- Check the md5 checksum and PGP signatures for any downloaded patches to verify that no Trojan horses have been introduced to the binaries.
- Change root passwords to be cryptic and unique, then use sudo (with your own account) for all administrative tasks.
- Remove compiler packages and do any needed compilations on a separate

machine. (This will make it harder for hackers to download and compile rootkits for their attacks.)

Security patches up to date

The current practice of monthly patching and weekly checks for security updates is sufficient.

Sensitive data is stored encrypted and how

Data is sent over the Internet encrypted

After disabling telnet and ftp, administrators should use SSH to connect to these servers. Also, SSL connections should be the only connections allowed to the administrative web sites for the iPlanet servers.

Anti-virus software is updated (if used as a server for Windows systems)

These servers do not interact with Windows systems directly.

Access is restricted to those with a need to know

The use of sudo will grant only the needed privileges to the various administrators in the organization as long as it is not used on utilities that can grant access to a shell.

The data center is located in a locked room with cipher locks on each of the 2 doors. The keypad has a shield over it to protect from others seeing the combination as it is entered. The combinations should be changed periodically.

The individual racks are not currently locked. They should be secured to prevent access to power/reset buttons as well as power and data connections.

Backup tapes are stored in the data center that is locked; however, you should consider putting these in a vault or locked cabinet that is only accessible by the backup operators.

Backup policies, disaster preparedness, etc.

Even with the suggested changes listed above, disaster plans need to be made in case of hardware failure or compromises to the system security. For security compromises, a disaster plan would include:

- Scripts to compare the local logs of each server to the logs stored on the syslog server.
- A bootable OS CD with all of the triage applications you would need to analyze a

system that has been compromised. This includes static linked binaries like ls, lsof, cd, ps, and all other basic system utilities. (without this, you may be troubleshooting a compromised system with compromised utilities.)

- Restore procedures to recover each server from backup tape. (Be sure to determine the last known secure state of the server, or you may be restoring from a backup that has already been compromised.)
- Backup tapes and servers are at the same physical location. Disasters like a fire would destroy all copies of this data.

Other issues/vulnerabilities

TARA is an updated version of the Tiger utility that will analyze your system and show possible security problems. It is up to you to determine the security/functionality tradeoffs of what is reported. Some white space has been removed to conserve space and comments are inserted in italics. It should be noted that the best way to run this utility is via tigercron. This allows you to schedule the scripts over time so it doesn't load down the system to much. It will also allow you to send email with only the changes since the last run.

First backup the tigerrc file and copy tigerrc-dist to tigerrc. The tigerrc-dist file enables all checks for the system.

```
# mv tigerrc tigerrc.bak
# cp tigerrc-dist tigerrc
# ./tiger
```

Output of TARA scripts ==========

```
Security scripts *** 2.0.9 ARC, 1999.0907.2100 ***
Wed Nov 22 10:45:56 EST 2000
10:45> Beginning security report for thumper (sun4u SunOS 5.8).

# Performing check of passwd files...
# Performing check of group files...
# Performing check of user accounts...
```

The accounts below should have their shells reset to insure no logins are possible. For each userid, use:

Usermod –s /bin/noshell <userid here>

This will not only deny access to the shell for these users, but will also log any attempts. You can get noshell from http://www.fish.com/security.

```
# Checking accounts from /etc/passwd.
--WARN-- [acc001w] Login ID adm is disabled, but still has a valid shell.
--WARN-- [acc005w] Login ID adm is disabled, but has a 'cron' file or cron
        entries.
--WARN-- [acc001w] Login ID bin is disabled, but still has a valid shell.
--WARN-- [acc001w] Login ID daemon is disabled, but still has a valid shell.
--WARN-- [acc001w] Login ID listen is disabled, but still has a valid shell.
--WARN-- [acc001w] Login ID lp is disabled, but still has a valid shell.
--WARN-- [acc005w] Login ID lp is disabled, but has a 'cron' file or cron
        entries.
--WARN-- [acc001w] Login ID noaccess is disabled, but still has a valid shell.
--WARN-- [acc001w] Login ID nobody4 is disabled, but still has a valid shell.
--WARN-- [acc001w] Login ID sys is disabled, but still has a valid shell.
--WARN-- [acc001w] Login ID uucp is disabled, but still has a valid shell.
--WARN-- [acc006w] Login ID adm's home directory (/var/adm) has group `sys'
        write access.
--WARN-- [acc006w] Login ID lp's home directory (/usr/spool/lp) has group `lp'
        write access.

# Performing check of /etc/hosts.equiv and .rhosts files...
# Checking accounts from /etc/passwd...
# Performing check of .netrc files...
# Checking accounts from /etc/passwd...
# Performing check of /etc/default/login, /securetty, and /etc/ttytab...
# Performing check of PATH components...
```

The following paths should be made read-only, removed from root's path, or checked with Tripwire routinely to prevent a user from replacing the executables with trojaned versions.

```
# Only checking user 'root'
--WARN-- [path001w] /usr/sbin/install.d in root's PATH from default is group
        `bin' writable.
--WARN-- [path001w] /usr/sbin/osa in root's PATH from default is group `bin'
        writable.
--WARN-- [path001w] /usr/bin/prodreg in root's PATH from default is group
        `root' writable.
--WARN-- [path001w] /usr/bin/prodregdir in root's PATH from default is group
        `root' writable.
--WARN-- [path001w] /usr/bin/rstartd in root's PATH from default is group
        `bin' writable.
# Performing check of anonymous FTP...
# Performing checks of mail aliases...
# Checking aliases from /etc/mail/aliases.
```

The following paths should be made read-only, removed from root's path, or checked with Tripwire routinely to prevent a user from replacing the executables with trojaned versions. Also, be sure to use full pathnames in cron jobs so you know which version is actually being executed.

```
# Performing check of `cron' entries...
--WARN-- [cron001w] cron entry for lp does not use full pathname:
--WARN-- [cron001w] cron entry for lp does not use full pathname:
--WARN-- [cron003] cron entry for root uses `/usr/lib/newsyslog' which
        contains `/usr/lib' which is group `bin' writable.
--WARN-- [cron003] cron entry for root uses `/usr/lib/fs/nfs/nfsfind' which
        contains `/usr/lib' which is group `bin' writable.
--WARN-- [cron001w] cron entry for root does not use full pathname:
--WARN-- [cron001w] cron entry for root does not use full pathname:
--WARN-- [cron003] cron entry for root uses `/usr/lib/gss/gsscred_clean' which
        contains `/usr/lib' which is group `bin' writable.
--WARN-- [cron003] cron entry for root uses `/usr/lib/gss/gsscred_clean' which
        contains `/usr/lib' which is group `bin' writable.
# Performing check of 'services' and 'inetd'...
# Checking services from /etc/services…
# Performing NFS exports check...
```

The following paths should be made read-only or checked with Tripwire routinely to prevent a user from replacing the executables with trojaned versions.

```
# Performing check of system file permissions...
--WARN-- [perm001w] /sbin should not have group write.
--WARN-- [perm001w] /usr/lib should not have group write.
--WARN-- [perm001w] /var should not have group write.
--WARN-- [perm021w] Disk device /dev/dsk/c0t0d0s0 has read access for group sys.
--WARN-- [perm021w] Disk device /dev/rdsk/c0t0d0s0 has read access for group sys.
--WARN-- [perm021w] Disk device /dev/dsk/c0t0d0s7 has read access for group sys.
--WARN-- [perm021w] Disk device /dev/rdsk/c0t0d0s7 has read access for group sys.
--WARN-- [perm021w] Disk device /dev/dsk/c1t5d0s2 has read access for group sys.
--WARN-- [perm021w] Disk device /dev/rdsk/c1t5d0s2 has read access for group sys.
--WARN-- [perm021w] Disk device /dev/dsk/c0t0d0s5 has read access for group sys.
--WARN-- [perm021w] Disk device /dev/rdsk/c0t0d0s5 has read access for group sys.
--WARN-- [perm021w] Disk device /dev/dsk/c0t0d0s3 has read access for group sys.
--WARN-- [perm021w] Disk device /dev/rdsk/c0t0d0s3 has read access for group sys.
--WARN-- [perm021w] Disk device /dev/dsk/c0t0d0s4 has read access for group sys.
--WARN-- [perm021w] Disk device /dev/rdsk/c0t0d0s4 has read access for group sys.
# Checking for known intrusion signs...
# Performing check of files in system mail spool...
# Performing system specific checks...
# Performing checks for SunOS/5...
```

The PROM needs to be secured with a password to keep hackers from permanently locking you out (which requires installing a new PROM chip!) Use the following command:

eeprom security-mode=command

--WARN-- [no-id] The PROM monitor is not in secure mode.
--WARN-- [misc008w] NFS port checking disabled in kernel.
Running './scripts/check_sendmail'...
Checking sendmail...

Remove the setuid permissions on any of the utilities listed below that are not needed at the user level.

Checking setuid executables...
--FAIL-- [fsys001f] File /etc/lp/alerts/printer is a setuid script:
-r-sr-xr-x 1 lp lp 203 Dec 16 1999 /etc/lp/alerts/printer
--WARN-- [fsys002w] setuid program /opt/ssh/bin/ssh1 has relative pathnames.
--WARN-- [fsys002w] setuid program /usr/bin/nispasswd has relative pathnames.
--WARN-- [fsys002w] setuid program /usr/bin/passwd has relative pathnames.
--WARN-- [fsys002w] setuid program /usr/bin/yppasswd has relative pathnames.
--WARN-- [fsys002w] setuid program /usr/lib/fs/ufs/ufsrestore has relative pathnames.
--WARN-- [fsys002w] setuid program /usr/openwin/bin/kcms_calibrate has relative pathnames.
--WARN-- [fsys002w] setuid program /usr/openwin/bin/kcms_configure has relative pathnames.
--WARN-- [fsys002w] setuid program /usr/openwin/bin/sparcv9/kcms_configure has relative pathnames.
--WARN-- [fsys002w] setuid program /usr/openwin/bin/sys-suspend has relative pathnames.
--WARN-- [suidxxx] Setuid file `/usr/openwin/bin/xlock' which is group `bin' writable.
--WARN-- [fsys002w] setuid program /usr/sbin/pgxconfig has relative pathnames.
--CONFIG-- [fsys003c] No setuid list... listing all setuid files
---s--x--x root bin /usr/lib/pt_chmod
-r-s--x--x root bin /usr/lib/lp/bin/netpr
-r-s--x--x root lp /usr/bin/cancel
-r-s--x--x root lp /usr/bin/lp
-r-s--x--x root lp /usr/bin/lpset
-r-s--x--x root lp /usr/bin/lpstat
-r-s--x--x root lp /usr/sbin/lpmove
-r-s--x--x root sys /usr/bin/admintool
-r-s--x--x uucp bin /usr/bin/tip
-r-sr-sr-x root bin /usr/openwin/bin/ff.core

```
-r-sr-sr-x root    daemon   /usr/dt/bin/sdtcm_convert
-r-sr-sr-x root    sys      /usr/bin/nispasswd
-r-sr-sr-x root    sys      /usr/bin/passwd
-r-sr-sr-x root    sys      /usr/bin/yppasswd
-r-sr-sr-x root    sys      /usr/dt/bin/dtaction
-r-sr-xr-x lp      lp       /etc/lp/alerts/printer
-r-sr-xr-x root    bin      /usr/bin/crontab
-r-sr-xr-x root    bin      /usr/bin/eject
-r-sr-xr-x root    bin      /usr/bin/fdformat
-r-sr-xr-x root    bin      /usr/bin/login
-r-sr-xr-x root    bin      /usr/bin/pfexec
-r-sr-xr-x root    bin      /usr/bin/rcp
-r-sr-xr-x root    bin      /usr/bin/rdist
-r-sr-xr-x root    bin      /usr/bin/rlogin
-r-sr-xr-x root    bin      /usr/bin/rmformat
-r-sr-xr-x root    bin      /usr/bin/rsh
-r-sr-xr-x root    bin      /usr/bin/sparcv7/uptime
-r-sr-xr-x root    bin      /usr/bin/sparcv7/w
-r-sr-xr-x root    bin      /usr/bin/sparcv9/uptime
-r-sr-xr-x root    bin      /usr/bin/sparcv9/w
-r-sr-xr-x root    bin      /usr/bin/volcheck
-r-sr-xr-x root    bin      /usr/bin/volrmmount
-r-sr-xr-x root    bin      /usr/dt/bin/dtappgather
-r-sr-xr-x root    bin      /usr/dt/bin/dtprintinfo
-r-sr-xr-x root    bin      /usr/dt/bin/dtsession
-r-sr-xr-x root    bin      /usr/lib/fs/ufs/quota
-r-sr-xr-x root    bin      /usr/lib/fs/ufs/ufsdump
-r-sr-xr-x root    bin      /usr/lib/fs/ufs/ufsrestore
-r-sr-xr-x root    bin      /usr/lib/sendmail
-r-sr-xr-x root    bin      /usr/lib/utmp_update
-r-sr-xr-x root    bin      /usr/sbin/pgxconfig
-r-sr-xr-x root    bin      /usr/sbin/ping
-r-sr-xr-x root    bin      /usr/sbin/pmconfig
-r-sr-xr-x root    bin      /usr/sbin/sparcv7/whodo
-r-sr-xr-x root    bin      /usr/sbin/sparcv9/whodo
-r-sr-xr-x root    bin      /usr/sbin/traceroute
-r-sr-xr-x root    sys      /usr/bin/chkey
-r-sr-xr-x root    sys      /usr/bin/sparcv7/ps
-r-sr-xr-x root    sys      /usr/bin/sparcv9/ps
-r-sr-xr-x root    sys      /usr/bin/su
-r-sr-xr-x root    sys      /usr/ucb/sparcv7/ps
-r-sr-xr-x root    sys      /usr/ucb/sparcv9/ps
-rws--x--x root    other    /opt/ssh/bin/ssh1
-rwsr-sr-x root    bin      /usr/openwin/bin/kcms_calibrate
-rwsr-sr-x root    bin      /usr/openwin/bin/kcms_configure
```

```
-rwsr-sr-x root     bin      /usr/openwin/bin/sparcv9/kcms_configure
-rwsr-xr-x root     bin      /usr/openwin/bin/sys-suspend
-rwsr-xr-x root     bin      /usr/openwin/lib/mkcookie
-rwsr-xr-x root     bin      /usr/sbin/allocate
-rwsr-xr-x root     bin      /usr/sbin/deallocate
-rwsr-xr-x root     bin      /usr/sbin/list_devices
-rwsr-xr-x root     bin      /usr/sbin/mkdevalloc
-rwsr-xr-x root     bin      /usr/sbin/mkdevmaps
-rwsr-xr-x root     sys      /usr/bin/at
-rwsr-xr-x root     sys      /usr/bin/atq
-rwsr-xr-x root     sys      /usr/bin/atrm
-rwsr-xr-x root     sys      /usr/bin/newgrp
-rwsr-xr-x root     sys      /usr/bin/newtask
-rwsr-xr-x root     sys      /usr/sbin/sacadm
-rwsrwxr-x root     bin      /usr/openwin/bin/xlock
```

END Output of tiger script ==========

Tripwire is a utility that makes a database of clean executables on your system. When run in check mode, it compares these initial values to the executables currently on the system and reports any changes.

To run Tripwire, you must first run:
tripwire –m I
to initialize the database, set the passphrase, generate the site and host keys, and encrypt the database.

After that, run:
tripwire –m c
to check for changes in the configuration. If you have just installed a package, or you know the changes are ok, you can run:
tripwire –m u
to update the database. (passphrase required)

Logcheck is a script that will look through your system logs for certain "interesting" entries and will email you if suspicious things are found. You can get it from http://www.sunfreeware.com.

Install the package with pkgadd and modify the following files so you only get mail when you want it:

logcheck.hacking – insert well known hacking messages here (causes alarm if found)
logcheck.violations – insert the "bad" search patterns here
logcheck.violations.ignore – insert specific errors to ignore here
logcheck.ignore – insert more general errors to ignore here

After editing the above files, just run the script and it will email you with any suspicious activity.

Example email with failed su attempts from logcheck.sh:

mail
From mdmiller Wed Nov 22 13:39:01 2000
Date: Wed, 22 Nov 2000 13:39:01 -0500 (EST)
From: Matt Miller <mdmiller>
Message-Id: <200011221839.NAA06908@thumper.fuqua.duke.edu>
Content-Length: 274

Security Violations
=-=-=-=-=-=-=-=-=-=-=
Nov 22 13:38:44 thumper su: [ID 810491 auth.crit] 'su root' failed for mdmiller on /dev/pts/2

Unusual System Events
=-=-=-=-=-=-=-=-=-=-=
Nov 22 13:38:44 thumper su: [ID 810491 auth.crit] 'su root' failed for mdmiller on /dev/pts/2

?