Global Information Assurance Certification Paper

Checklist For Installing A Secure Irix 6.5 Workstation

GARY SAMUEL

This is an account of the steps that an administrator should take to secure an SGI Indigo2 workstation from out-of-the-box to an "Internet ready" state. The specifications of the system are as follows:

System:	SGI Indigo2
Hardware:	MIPS IP22, 250 MHz, R4400 processor
	192 MB RAM
	4 GB SCSI system drive
	4 GB SCSI option drive
	SCSI CD-ROM
Operating System:	SGI IRIX 6.5.10

This machine will be installed as a developer's workstation capable of developing and compiling software using both SGI's native MIPSpro compilers and the GNU compilers. Other system requirements are:

- Access to the Internet for the purpose of general web browsing etc.
- Access other hosts on the company's network for the purpose of shell sessions and file transfers.
- Ability to send email.
- Function as a DNS client.
- Do not use NFS, NIS or RPC service.

Command (PROM) Monitor

To get into the PROM on an SGI system, turn the system on and use the mouse to click the System Maintenance icon displayed during the boot sequence. Enter the Command Monitor by clicking the similarly labeled icon. On machines without graphics capability a text menu maps to the numeric keys.

At the Command Monitor prompt (>>), issue the "**passwd**" command and set your PROM password. Please ensure that you enter the password properly and remember it since the only way to reset a forgotten PROM password is to remove a jumper on the CPU main board.

Setting this password provides a measure of physical security, as the machine will not boot to miniroot from CD-ROM or perform any Command Monitor mode commands and diagnostics without it. While in the Command Monitor, type "**printenv**" and ensure that the AutoLoad variable is set to yes, the OSLoadPartition variable is set to scsi(0)disk(1)rdisk(0)partition(0) and the OSLoadFilename is /unix

PROM Checklist

1. ____ Password set with passwd command
2. ____ AutoLoad variable set to yes
3. ____ OSLoadPartition variable set to root partition of the first SCSI disk.
4. ____ OSLoadFilename variable set to /unix (the kernel)

IRIX 6.5.10 Media
As of the writing of this document, IRIX 6.5.10 is the latest Quarterly Update released by SGI. This release contains updates for both the maintenance and feature streams. Approximately 156 customer reported bugs are fixed in this release. IRIX 6.5.11 is in development and planned for release late February 2001.

The 6.5.10 update consists of 4 CD-ROMs;

- IRIX Installation Tools and Overlays [1 of 3]
- IRIX Overlays [2 of 3]
- IRIX Overlays [3 of 3]
- IRIX Applications

In addition, the following CD-ROMs from the IRIX 6.5 base release will also be needed.

- IRIX 6.5 Installation (original dated June 1998)
- IRIX 6.5 Foundation 1
- IRIX 6.5 Foundation 2
- IRIX 6.5 Development Foundation
- IRIX 6.5 Development Libraries

OS CD-ROMs checklist

1. _____ Do you have all the necessary CD-ROMs?

Partition the File System
Load the original 6.5 Installation tools CD into the CD-ROM drive and boot the machine. Enter the Command Monitor mode, type the "**hinv**" command and make note of the SCSI ID and controller of the CD-ROM drive. Using that information, type the following command. (My examples below assume the CD-ROM SCSI ID is 3, controller is 0 and the disk SCSI ID is 1 and the controller is 0)

boot –f dksc(0,3,8)sashARCS dksc(0,3,7)stand/fx.ARCS --x

This command (specific for the IP22, IP20 and IP19 CPUs) loads the stand-alone shell and starts up the format application in the expert mode. Please see SGI documentation for other CPU types. Rather than use the single root partition that IRIX creates by default, partition the file system as follows:

Partition 0 = / _____ MB
Partition 1 = swap _____ MB (at least as much as RAM)
Partition 4 = /var _____ MB
Partition 6 = /usr _____ MB

After partitioning, exit the Command Monitor and load the 6.5.10 Installation CD. Select Install System Software menu item and boot to miniroot. As the machine boots to miniroot it automatically creates the /, swap and /usr file systems because the 0, 1 and 6

partitions are so defined by IRIX. The admin subprogram must be used to create and mount the /var file system.

Partitions Checklist

1. ____ At the Inst> prompt type "**13"** to get to the Admin> prompt.
2. ____ At the Admin> prompt type "**mkfs /dev/dsk/dks0d1s4**" to create the file system.
3. ____ At the Admin> prompt type "**sh**" to get a shell prompt.
4. ____ At the # prompt type "**mkdir /root/var**" to create a mount point.
5. ____ At the # prompt type "**mount /dev/dsk/dks0d1s4 /root/var**" to mount it.

Installing IRIX 6.5.10

Exit up to the Inst menu and load each of the OS CDs in the order listed below.

- IRIX 6.5.10 Installation Tools 1
- IRIX 6.5.10 Installation Tools 2
- IRIX 6.5.10 Installation Tools 3
- IRIX Applications
- IRIX Foundation 1
- IRIX Foundation 2
- IRIX Development Foundation
- IRIX Development Libraries

After loading the last CD, select the software subsystems to install. SGI selects several unnecessary products as part of its default/standard installation package. These should not be installed.

Installation Checklist

1. ____ At the Inst> prompt type "**from /CDROM/dist**" and continue to load CDs.
2. ____ After the last CD, type "**done**".
3. ____ At the Inst> prompt type "**keep ***"
4. ____ At the Inst prompt type "**install standard**"
5. ____ At the Inst> prompt type "**keep <items in do_not_install list>**" See Appendix A
6. ____ At the Inst> prompt type "**install eoe.sw.acct eoe.sw.audit**" to install the auditing and accounting software that are not loaded by default.
7. ____ At the Inst> prompt type "**go**" and replace CDs as requested by the program.

Following the completion of the install, use the admin submenu to edit the /etc/fstab file so that the /var/partition will be mounted at boot time.

8. ____ Add the following line to /etc/fstab:
   /dev/dsk/dks0d1s4 /var xfs rw,raw=/dev/rdsk/dks0d1s4 0 0

Exit Inst and allow the machine to reboot as prompted by the install program. The machine reboots to an "icon login screen" which displays icons for all current users of the

system. This is somewhat unnerving for an administrator and will be fixed later on in the process.

Installing Additional Software

SGI maintains a freeware distribution on the web at http://freeware.sgi.com. It contains useful third-party packages such as gcc, openssh, tripwire and many others. They are all in SGI's instable format and provide the option of downloading and installing as opposed to building them. Since this machine will be a developer's system, I suggest installing just the gcc compiler and use it to build all the additional packages. Administrators who build software can manipulate just about anything and set whatever features they wish rather than having to guess what the original builder of the SGI freeware did!
A good idea is to burn a CD containing the gcc software in tardist format. Place the CD in the CD-ROM drive.

Additional Software Checklist

1. ___ **mkdir /tmp/sware**
2. ___ **cd /tmp/sware**
3. ___ **cp /CDROM/fw_gcc-2.95.2-sgipl1.tardist .**
4. ___ **tar xvf fw_gcc-2.95.2-sgipl1.tardist**
5. ___ **inst –f .**

This installs a fully functional g++/gcc compiler suite in /usr/freeware, which we will later use to build and install the following security related packages:

- Sendmail
- Network Time Protocol (xntp)
- TCP Wrappers
- Tripwire
- OpenSSL
- OpenSSH

Check and Install Current Recommended Patches

SGI also maintains a security advisory/patches website at http://www.sgi.com/support/security. Check advisories and download any 6.5.10 patches for installation on the system. Administrators should also get regular security updates from a variety of sources including CERT and SANS, which keep a very close watch on OS vulnerabilities.
The current advisory (at the time of this writing) contains two possible security concerns.

- InPerson, which was not install.
- Telnetd daemon that is installed by default, but is not vulnerable in 6.5.10. In any event telnetd will be disabled later on.

Patches Checklist

1. ___ Check current advisories at website
2. ___ Create CD with necessary patches/fixes for IRIX OS and applications.
3. ___ Install if necessary.

Configure Administrative and Users Accounts

Accounts Checklist

1. ___ Type "**passwd**" to assign root password.
2. ___ Do not allow direct root logins. To improve the root auditing process, all direct root logins should be denied. A user should be forced to login as a non-privileged user, then use **/bin/su** to gain root access. Before this is done, a regular user account should be created.
3. ___ Add a new non-privileged user.
4. ___ Open /etc/default/login and look for the line **CONSLOE=/dev/CONSOLE**. It will be commented out. Uncomment it and set it to read **CONSOLE=""**.
5. ___ Also in /etc/default/login, find the line **MANPASS=NO** and change it to **MANPASS=YES**. This causes the system to lock out each account that does not have a password.
6. ___ To ensure that root does not inadvertently execute Trojan horse programs placed by hackers, root's PATH environmental variable should not include the current directory (.). Check root's .cshrc, .login and .profile to verify this.
7. ___ To guard against race conditions exploits, make sure that there are no world writable directories in root's PATH. Check root's .cshrc and .profile to verify this.
8. ___ Set restrictive umask for all users and an even more restrictive one for root. In /etc/default/login find the line **UMASK=<###>** and ensure that it is set to **UMASK=022**. Edit root's .cshrc and .profile and set **umask 027**. Also set **umask 022** in /etc/profile and /etc/cshrc. All files created by root will now have rw-r----- and directories will have rwxr-x--- permissions. Any special needs can be addressed on an individual basis.
9. ___ Check root's .cshrc and .profile to ensure that root does not source non root-owned files or files that are group and world writable at login.

A newly installed SGI has several accounts with no password and these should be locked. In addition there are a number of special accounts whose home directories do not exist and should also be disabled by locking.

1. ___ Issue **/usr/sbin/pwck** which checks the /etc/passwd file for inconsistencies. It should report that the sysadm, cmwlogin, nuucp, auditor, dbadmin, rfndd, demos, OutOfBox and 4Dgifts accounts do not have home directories. Proceed to lock them.
2. ___ Issue **/usr/bin/passwd –l <account name>** for all of the above accounts as well as the lp, Ezsetup and guest accounts which do not have password.
3. ___ Edit /etc/passwd to assign each disabled account an invalid shell by making /dev/null the default shell.

Earlier we said that by default IRIX presents an "icon login screen" containing an icon for each valid user account on the system. For obvious security reason it is not a good idea to announce a list of accounts, so this feature should be turned off.

4. ___ **/sbin/chkconfig** noiconlogin on
5. ___ **/sbin /chkconfig** visuallogin off

Every entry in the /etc/passwd file should now be a valid account with a password or a locked account with an invalid login shell. Now convert to the shadow password system.

6. ___ **/sbin/pwconv**

File System Setup

Earlier the /, /usr and /var partitions were defined. To prevent setuid scripts from executing on /var mount it with the nosuid switch. To prevent hackers from installing Trojan horse replacement of system binaries in /usr mount it with the read-only switch.

File System Checklist

1. ___ Edit the /etc/fstab file to replace rw with ro for the /usr entry.
2. ___ Edit the /etc/fstab file to add the nosuid option for the /var entry.

These entries should look like this:

/dev/usr /usr xfs ro,raw=/dev/rusr 0 0

/dev/dsk/dks0d1s4 /var xfs nosuid,rw,raw=/dev/rdsk/dks0d1s4 0 0

N.B. While we are installing the system, mount /usr read-write. Revert to read-only when finished.

Customizing Boot Services

IRIX provides a means to turn individual services on or off at boot time. Chkconfig is a configuration state checker which examines configuration flags found in the /etc/config file. If a flag is on, the service will be started at boot time. Since the ONC/NFS software is not installed, there are no flags for nfsd, biod, cachefs or autofs. Also NIS is not to be used, so the nsd service (which has replaced ypserv and ypbind) will not be started. These are the services that should be turned off.

Boot Services Checklist

1. ___ **/sbin/chkconfig nsd off** Turns off the name service daemon
2. ___ **/sbin/chkconfig ipaliases off** Turns off multiple network interface support
3. ___ **/sbin/chkconfig esp off** Turns off SGI embedded support
4. ___ **/sbin/chkconfig timed off** BSD time syncro daemon. We will use NTP
5. ___ **/sbin/chkconfig sendmail off** We will build and install nullclient version
6. ___ **/sbin/chkconfig sendmail_cf off** Do not generate sendmail config file

We should also create a script to set the default umask for system processes. **[1]**

1. ___ **echo 'umask 022' > /etc/init.d/umask.sh**
2. ___ **/sbin/chmod 744 /etc/init.d/umask.sh**
3. ___ **/sbin/ln –s /etc/init.d/umask.sh /etc/rc0.d/S00umask.sh**
4. ___ **/sbin/ln –s /etc/init.d/umask.sh /etc/rc2.d/S00umask.sh**

Streamline /etc/inetd.conf

Inetd allows a lot of insecure and unnecessary services by default. Some of the services started by inetd are ftp, telnet, shell, login, exec, finger, http, wn_http, bootp, tftp, tcpmux, echo, chargen, sgi_videod and sgi_fam. When pruning is finished, only day-to-day services necessary for IRIX should remain. The only really necessary service is sgi_fam (the File Alteration Monitor). This server tracks changes to the filesystem and relays them to applications such as the file manager and mailbox. When fam is run from inetd, a security weakness is introduced where it is possible for rouge clients to obtain

names of all the files and directories on the system. Create a script to run fam in the local_only mode and modify the /etc/config/inetd.options file so that inetd just checks for the config file, but does not startup at boot time.

Inetd Checklist

1. ___ **/sbin/rm /etc/inetd.conf /usr/etc/inetd.conf**
2. ___ **echo '-s' > /etc/config/inetd.options**
3. ___ Edit /etc/fam.conf to replace the line "local_only = false" with "local_only = true".
4. ___ **/sbin/touch /etc/init.d/network.local**
5. ___ **/sbin/ln –s /etc/init.d/network.local /etc/rc2.d/S31network**
6. ___ Edit /etc/init.d/network.local to add the following script:

```
#!/bin/sh
# Local networking things
case $1 in
  'start')
       if [ -x /usr/etc/fam –a –f /etc/fam.conf ] ; then
           /usr/etc/fam –c /etc/fam.conf
       fi
       ;;
  'stop')
        killall fam
       ;;
   *)
        echo "Usage: $0 {start | stop}"
      ;;
esac
exit 0
```

7. ___ **/sbin/chmod 744 /etc/init.d/network.local**
8. ___ **/etc/reboot**

Setup Networking and Hostname Resolution

Networking Checklist

Create the /etc/resolv.conf

1. ___ **/sbin/touch /etc/resolv.conf**
2. ___ **echo 'domain <DOMAINNAME>' > /etc/resolv.conf**
3. ___ **echo 'nameserver <IP ADDRESS OF NAMESERVER>' >> /etc/resolv.conf**
4. ___ **/sbin/chmod 644 /etc/resolv.conf**

Edit /etc/nsswitch.conf, /etc/sys_id, /etc/hosts and other network files.

1. ___ Edit /etc/nsswitch.conf to replace the line "hosts: nis dns files" with "hosts: files dns". Remove all other occurrences of the word "nis" from this file.
2. ___ **echo '$ROUTE $QUIET add net default <IP ADDRESS OF GATEWAY>' >> / etc/config/static-route.options**
3. ___ **echo '<HOSTNAME>' > /etc/sys_id**

4. ___ **echo '< IP ADDRESS> <HOSTNAME.DOMAINNAME> <ALIAS>' >> /etc/hosts**
5. ___ Edit /etc/TIMEZONE to indicate the correct time zone for your location.

Miscellaneous Network and System Modifications

Cron

The root crontab uses a bad umask (033) which causes /var/adm/SYSLOG and other log files to be created with group and world read permissions.

1. ___ Edit /var/spool/cron/crontab/root to replace each 033 entry with 077.
2. ___ Ensure that the /var/spool/cron/crontab directory contains entries for root and sys alone. Remove all others.

Statutory Warnings

Post your site's custom statutory warning message in the /etc/motd and /etc/issue files.

1. ___ Edit /etc/motd and /etc/issue to include warning message.

Kernel Parameters

Core dumps are generally world readable. Hackers can cause them to be generated and then read data such as the /etc/shadow file from them. They can also be used in denial of service attacks. The rlimit_core_max kernel parameter specifies the maximum size of a core file and is set to a large value by default. Setting this value to 0 will restrict the generation of core files. This is only a small inconvenience to developers who can still use tools such as CaseVision Tools and Insure++ for debugging.

1. ___ **/usr/sbin/systune rlimit_core_max 0**

By default the kernel parameter restricted_chown is set to 0, which allows users to giveaway file ownership System V style. This is a security risk that has resulted in several recent exploits. Change this value to 1 to enforce the BSD style chown, which only allows root to give away files.

2. ___ **/usr/sbin/systune restricted_chown 1**

Disable ipforwarding to prevent broadcasting of sensitive system information.

3. ___ **/usr/sbin/systune ipforwarding 0**

Disable ipsendredirects

4. ___ **/usr/sbin/systune ipsendredirects 0**

Disable ipdirected_broadcast

5. ___ **/usr/sbin/systune ipdirected_broadcast 0**

Reconfigure the kernel and reboot

6. ___ **/etc/autoconfig**
7. ___ **/etc/reboot**

The nfs_portman parameter is not an issue here since the NFS subsystem is not installed. Note also that there is no kernel parameter in the current IRIX 6.5.10 kernel for

preventing execution of the stack. As a result of my research, this issue was raised with SGI who has since initiated a RFE (request for enhancement) to solve this in a future upgrade release of the 0S.

Robust System Logging

As initially configured, IRIX does not log authorization information to /var/adm/SYSLOG via syslogd. Su attempts are logged in /var/adm/sulog, but other important information such as failed login, xdm, ssh, getty, ftpd and rshd attempts are not. To setup authorization logging;

1. ___ Edit /etc/syslog.conf to add the line auth.info <TAB> /var/adm/authlog
2. ___ **/sbin/touch /var/adm/authlog**
3. ___ **/sbin/chown root:sys /var/adm/authlog**
4. ___ **/sbin/chmod 600 /var/adm/authlog**

Also log unsuccessful login attempts at the console.

1. ___ **/sbin/touch /var/adm/loginlog**
2. ___ **/sbin/chown root:sys /var/adm/loginlog**
3. ___ **/sbin/chmod 600 /var/adm/loginlog**

IRIX includes log rotation as part of the /var/spool/cron/crontabs/root cron file.

1. ___ Edit the file to add a line similar to the one below for each new log file.

Rotate the logs
1 1 * * 0 umask 077; cd /var/adm; if test –s authlog && test "'/sbin/stat –qs authlog'" –ge 10240; then mv –f authlog OLDauthlog; touch authlog; killall 1 syslogd; fi

Setup Process Accounting

IRIX process accounting provides information the administrator can use to determine resource usage and track system events on a process-by-process basis. Both of these functions are critical if ever there is a need for computer forensics. The /var/adm/pact file (to which the kernel writes its information) can grow quickly depending upon the system usage, so when accounting is on the size of the /var partition should be monitored closely. If space is limited, turn accounting on to gather specific data only if something suspicious warrants it.

To turn accounting on:

Ensure that the accounting subsystem is installed.

1. ___ **/usr/sbin/versions | grep eoe.sw.acct**

Turn on the chkconfig flag for startup at boot time.

2. ___ **/sbin/chkconfig acct on**

Start the kernel writing to /var/adm/pact

3. ___ **/usr/lib/acct/startup**

To turn accounting off:

4. ___ **/sbin/chkconfig acct off**
5. ___ **/usr/lib/acct/shutacct**

Setup The System Audit Trail

The IRIX auditing system features a number of commands and switches which allow the administrator to review all system activities such as trends in system usage, unsuccessful attempts to use system resources, attempts at guessing root password and attempts to access files owned by other users. Because auditing has the potential to use large amounts of disk space, It is recommended that it be implemented on a "need" basis on systems where disk space is limited.

To turn auditing on:

Ensure that the eoe.sw.audit subsystem is installed.

1. ___ **/usr/sbin/versions | grep eoe.sw.audit**
2. ___ **/sbin/chkconfig audit on**
3. ___ **/etc/init.d/audit start**

IRIX has a preconfigured auditing environment and begins to write auditing information on these events immediately. See Appendix B **[2]** for a list of events IRIX audits by default.

To turn auditing off:

1. ___ **/sbin/chkconfig audit off**
2. ___ **/etc/init.d/audit stop**

Building and Installing Sendmail

By default, IRIX installs sendmail 8.9.3 as part of the eoe.sw.base software subsystem. Although this is secure, build and install the latest version which includes several bug fixes. There will be no local delivery of mail to this machine therefore the sendmail daemon should not be started. Configure sendmail to run in the "nullclient" mode where all mail is forwarded to a central mailhub for delivery.

Sendmail Checklist

1. ___ Get the 8.11.0 source from ftp://ftp.sendmail.org/pub/sendmail/sendmail.8.11.0.tar.gz
2. ___ **/usr/sbin/gzcat sendmail.8.11.0.tar.gz | tar xvf –**
3. ___ **cd sendmail-8.11.0**
4. ___ Read the README files in the toplevel, devtools, sendmail and cf directories.
5. ___ If you wish to use gcc to perform the build then edit the "IRIX.6.5" file in the devtools/OS directory and replace the line "define confCC cc" with "define confCC gcc".
6. ___ **/bin/sh Build**
7. ___ After the build is successfully completed, you will have a sendmail binary in the obj.IRIX.6.5.IP22/sendmail directory.
8. ___ **cd cf/cf**
9. ___ **cp clientproto.mc <HOSTNAME>.mc**

10. ___ Edit <HOSTNAME>.mc to include the OSTYPE and FEATURE lines showing the OS and the name of the mailhub to which mail should be sent. Here is what those two lines in this file should look like:

OSTYPE(IRIX6)
FEATURE(nullclient, <MAILHUB>.$m)

Note that <MAILHUB> is the hostname of the mailhub on your local network.

11. ___ **/sbin/m4 ../m4/cf.m4 <MAILHUB>.mc > sendmail.cf**
12. ___ **cp sendmail.cf /etc/mail**
13. ___ **/sbin/chown root:sys /etc/mail/sendmail.cf**
14. ___ **/sbin/chmod 644 /etc/mail/sendmail.cf**
15. ___ **/sbin/ln –s /etc/mail/sendmail.cf /etc/sendmail.cf**
16. ___ **cd ../../obj.IRIX.6.5.IP22/sendmail**
17. ___ **cp sendmail /usr/lib** [replacing IRIX,s binary]
18. ___ **/sbin/chown root:sys /usr/lib/sendmail**
19. ___ **/sbin/chmod 4555 /usr/lib/sendmail**

This completes the installation of the daemon and configuration file. Configure the startup environment so that the sendmail daemon is not started at boot time.

20. ___ **/sbin/chkconfig sendmail off**

Since sendmail is not running in daemon mode, there is a chance that mail that is not forwarded immediately to the mailhub (due to unavailability or network congestion) may remain in the queue indefinitely. To solve this problem, set up a cron job to flush the mail queue intermittently. The following line when added to the /var/spool/cron/crontabs/root file, will flush the queue every fifteen minutes:

21. ___ **0,15,30,45 * * * * /usr/lib/sendmail –q > /dev/null 2>&1**
22. ___ **kill –HUP <cron PID>**

Building and Installing TCP Wrappers

TCP Wrappers Checklist

1. ___ Get the source from ftp://ftp.porcupine.org/pub/security/tcp_wrappers_7.6.tar.gz
2. ___ **/usr/sbin/gzcat tcp_wrappers_7.6.tar.gz | tar xvf –**
3. ___ **cd tcp_wrappers_7.6**
4. ___ Read the README.IRIX document.
5. ___ **/sbin/chmod 644 Makefile**
6. ___ Edit Makefile to include the following lines:

```
# IRIX
  REAL_DAEMON_DIR = /usr/sbin
  CC = /usr/freeware/bin/gcc  #The location of gcc
  FACILITY = LOG_AUTH
```

7. ___ **/sbin/make irix6**

This should have successfully built the software and produced related binaries, header and include files in the tcp_wrappers_7.6 directory. Proceed to install the software.

8. ___ **/sbin/mkdir –p /usr/local/sbin /usr/local/include**
9. ___ **cp safe_finger tcpd tcpdchk tcpmatch try-from /usr/local/sbin**
10. ___ **/sbin/chmod 0555 /usr/local/sbin/***
11. ___ **/sbin/chown root:daemon /usr/sbin/***
12. ___ **cp libwrap.a /usr/local/lib** [directory exists by default]
13. ___ **/sbin/chown root:daemon /usr/local/lib/libwrap.a**
14. ___ **/sbin/chmod 0555 /usr/local/lib/libwrap.a**
15. ___ **cp tcpd.h /usr/local/include**
16. ___ **/sbin/chown root:daemon /usr/local/include/tcpd.h**
17. ___ **/sbin/chmod 0444 /usr/local/include/tcpd.h**

Secure Shell Requirements

Because rlogin, telnet, ftp and other clear text protocols were disabled, install SSH as the means of remotely accessing the system and transferring file to and from it securely. Build and install OpenSSH 2.5.1p1, which is the latest opensource version as of this writing. To successfully build it, working installations of OpenSSL and the Zlib compression library are needed.

Building OpenSSL

1. ___ Download openssl-0.9.6.tar.gz from http://www.openssl.org
2. ___ **/usr/sbin/gzcat openssl-0.9.6.tar.gz | tar xvf –**
3. ___ **cd openssl-0.9.6**
4. ___ Read INSTALL file and note the requirements. Note Perl5 which is installed by default at /usr/sbin/perl.
5. ___ **./Configure irix-gcc**
6. ___ Edit Makefile to set the proper location of gcc.
7. ___ **/sbin/make**
8. ___ **/sbin/make test**
9. ___ **/sbin/make install** [Installs the software to /usr/local/ssl]

Building zlib

1. ___ Download zlib.tar.gz from http://www.freesoft.com/pub/infozip/zlib
2. ___ **/usr/sbin/gzcat zlib.tar.gz |tar xvf –**
3. ___ **cd zlib-1.1.3**
4. ___ **setenv CC gcc; ./configure**
5. ___ **/sbin/make test**
6. ___ **/sbin/make install**

Building and Installing SSH

Having satisfied the prerequisites, proceed to build OpenSSH.

1. ___ Download openssh-2.5.1p1.tar.gz from http://www.openssh.com
2. ___ **/usr/sbin/gzcat openssh-2.5.1p1.tar.gz | tar xvf –**

3. ___ **cd openssh-2.5.1p1**
4. ___ **setenv CFLAGS –I/usr/local/include**
5. ___ **setenv LDFLAGS –L/usr/local/lib**
6. ___ **./configure --with-tcp-wrappers --sysconfdir=/etc/ssh --without-rsh --disable-suid-ssh**
7. ___ **/sbin/make**
8. ___ **/sbin/make install**

The sshd daemon is now installed in /usr/local/sbin and ssh, scp, sftp, ssh-keygen, etc are installed at /usr/local/bin. The ssh_config, sshd_config and public/private key files are installed in /etc/ssh.

Configuring Secure Shell with TCP Wrappers

Because SSH was built with tcp-wrappers support, all that needs to be done is to construct the /etc/hosts.allow and /etc/hosts.deny files.

1. ___ **/sbin/touch /etc/hosts.allow**
2. ___ **/sbin/chown root:sys /etc/hosts.allow**
3. ___ **/sbin/chmod 600 /etc/hosts.allow**
4. ___ **echo 'ssh: <IP ADDRESS>' > /etc/hosts.allow**
5. ___ **echo '# Comment: This entry added <DATE> for <NAME>' >> /etc/hosts.allow**. For auditing purposes it is always a good idea to document reasons for entries in this file.
6. ___ **/sbin/touch /etc/hosts.deny**
7. ___ **/sbin/chown root:sys /etc/hosts.deny**
8. ___ **/sbin/chmod 600 /etc/hosts.deny**
9. ___ **echo 'ALL: ALL: /usr/bin/mail -s % d-% a root' > /etc/hosts.deny**

%d refers to the application name and %a refers to the client's IP address. To further explain refusal or to snap a warning back at the offending client, the banner option can be included in the line. The line in the deny file would then read:
ALL: ALL: /usr/bin/mail -s % d-% a root: banners /etc/banners/deny
Create the /etc/banners/deny directory with a message file named sshd.

Configuring SSHD

To prepare the machine to accept connections from ssh clients, the /etc/ssh/sshd_config file must be modified.
Edit /etc/ssh/sshd_config as follows:

1. ___ X11Forwarding yes
2. ___ PermitRootLogin no
3. ___ PrintMotd no
4. ___ KeepAlive no

To start sshd at boot time create a script.

5. ___ Save the following script as /etc/init.d/sshd. **[3]**

```
#!/bin/sh
case $1 in
  'start')
```

```
        if [ -x /usr/local/sbin/sshd –a –f /etc/ssh/sshd_config ] ; then
                /usr/local/sbin/sshd –f /etc/ssh/sshd_config
        fi
        ;;
'stop')
        kill `cat /etc/ssh/sshd.pid`
        ;;
  *)
        echo "Usage: $0 {start : stop}"
        ;;
esac
exit 0
```

6. ___ **/sbin/chown root:sys /etc/init.d/sshd**
7. ___ **/sbin/chmod 744 /etc/init.d/sshd**
8. ___ **ln –s /etc/init.d/sshd /etc/rc2.d/S30sshd**
9. ___ **/etc/init.d/sshd start**

Remote clients listed in the /etc/hosts.allow file can now connect to this machine.

Building and Configuring XNTP

Having the machine display the correct date and time at all occasions is extremely important. This feature is useful during forensic investigations where time correlation is critical in determining the flow of data between machines. XNTP is an implementation of the Network Time Protocol that synchronizes time between computers on the Internet.

1. ___ Download xntp3-5.93.tar.gz from http://www.eecis.udel.edu/~ntp
2. ___ **/usr/sbin/gzcat xntp3-5.93.tar.gz |tar xvf –**
3. ___ **cd xntp3-5.93**
4. ___ Read the README and INSTALL files
5. ___ **./configure**
6. ___ **/sbin/make**
7. ___ **/sbin/make check**
8. ___ **/sbin/make install** [Installs the package in /usr/local/bin]

Next build the /etc/ntp.conf file which xntpd reads at startup.

9. ___ Select three stratum 2 time servers from the public NTP server list at http://www.eecis.udel.edu/~mills/ntp/clock2.htm.
10. ___ Contact the administrator for each and request permission to suck clock form them. It is important to ask because their servers may already be heavily loaded.
11. ___ Add the lines to /etc/ntp.conf. Most servers use DNS aliases for real machine names. It is recommended that the alias be used rather than the real name to prevent disconnection in the event that the clock is moved to another machine later on. Here is the format of the /etc/ntp.conf file:

```
server<ALIAS>
server<ALIAS>
server<ALIAS>
driftfile       /etc/ntp.drift
```

12. ___ **/sbin/chkconfig timed off** [Turn off IRIX timed daemon]
13. ___ **/sbin/touch /etc/init.d/xntp**
    Edit the file to add the following script: **[4]**

```
#!/bin/sh
NTPDATE=/usr/local/bin/ntpdate
XNTP=/usr/local/bin/xntpd
case $1 in
    'start')
        if test –x $NTPDATE; then
          $NTPDATE –s <ALIAS>
        fi
        if test –x $XNTP; then
          $XNTP && echo `Starting xntp`
        fi
        ;;
    'stop')
        /sbin/killall –k 10 –TERM xntpd
        ;;
    *)
        echo "Usage: /etc/init.d/xntp {start | stop}"
        ;;
  esac
```

14. ___ **/sbin/chmod 744 /etc/init.d/xntp**
15. ___ **/sbin/chown root:sys /etc/init.d/xntp**
16. ___ **ln –s /etc/init.d/xntp /etc/rc3.d/S90xntp**
17. ___ **ln –s /etc/init.d/xntp /etc/rc2.d/K90xntp**
18. ___ **/etc/init.d/xntp start** [startup the xntp daemon]

Building and Configuring Tripwire

Tripwire is a host-based Intrusion Detection System (IDS) that uses cryptographic signatures of key system files to determine unauthorized modification or access. Tripwire is most effective if installed just after the machine is fully prepared and just before it is ready to be integrated into the network.

1. ___ Download Tripwire-1.3.1-1.tar.gz from http://www.tripwiresecurity.com
2. ___ **/usr/sbin/gzcat Tripwire-1.3.1-1.tar.gz | tar xvf –**
3. ___ **cd tw_ASR_1.3.1_src**
4. ___ Edit include/config.h and set the CONFIG_PATH variable to /usr/tripwire and DATABASE_PATH to /usr/tripwire/databases.

Remember that /usr is our read-only file system, which has been temporarily configured read-write for software installation.

5. ___ **/sbin/make**
6. ___ **/sbin/make test**
7. ___ **cp src/tripwire /usr/local/bin**
8. ___ **cp scr/siggen /usr/local/bin**
9. ___ **/sbin/chown root:sys /usr/local/bin/tripwire**
10. ___ **/sbin/chmod 0500 /usr/local/bin/tripwire**
11. ___ **cd configs**
12. ___ **cp tw.conf.irix /usr/tripwire/tw.config**
13. ___ **/sbin/chmod 600 /usr/tripwire/tw.config**
14. ___ The administrator should customize this file to suit the software installation at her site, as well as include all setuid and setgid files. However, this default configuration file provides checking for system configuration and binary files.
15. ___ **cd /usr/tripwire**
16. ___ **/usr/local/bin/tripwire –initialize** [To create the database file at $DATABASE_PATH specified in the config.h file]
17. ___ Add the following line to /var/spool/cron/crontabs/root to automate an integrity check at 0500 hrs daily and mail root the results:

0 5 * * * if test –x /usr/local/bin/tripwire; then /usr/local/bin/tripwire 2>1& | /usr/sbin/Mail –s Tripwire_result root; fi

Data Integrity and Recovery

Two additional steps should be taken to ensure data integrity and data recovery in case of system breech or failure.

Clone System Disk

After installing all software, the system disk should be cloned.

1. ___ Install a second disk of similar size to the system disk, bring the machine up and execute "**hinv**" to note SCSI ID and controller information.
2. ___ Partition the second disk in the same way as the system disk, i.e. with /, /usr and /var partitions similarly sized.
3. ___ Use **/sbin/dvhtool** to copy the contents of the volume header of the system disk as well as the fx and sash programs to the new disk. See Appendix C for details on how this is done. **[5]**
4. ___ Make new filesystems on each partition of the new disk using /sbin/mkfs. To make the new filesystem on the /, /usr and /var partitions of a disk with SCSI ID 2 on controller 0, the commands would be:
5. ___ **/sbin/mkfs /dev/dsk/dks0d2s0**
6. ___ **/sbin/mkfs /dev/dsk/dks0d2s6**
7. ___ **/sbin/mkfs /dev/dsk/dks0d2s4**
8. ___ **/sbin/mkdir /clone** [Create a temporary mount point for the new disk]
9. ___ **/sbin/mount /dev/dsk/dks0d2s0 /clone** [Mount the root partition at /clone]
10. ___ **cd /clone**

11. ___ **/usr/sbin/xfsdump –l 0 - / | /sbin/xfsrestore - .** [Level 0 dump of / to new disk]
12. ___ **cd ..**
13. ___ **/sbin/umount /clone**
14. ___ **/sbin/mount /dev/dsk/dks0d2s6 /clone**
15. ___ **cd /clone**
16. ___ **/usr/sbin/xfsdump –l 0 - /usr | /sbin/xfsrestore - .** [Dump /usr to new disk]
17. ___ Repeat procedure for /var partition.
18. ___ **cd ..** [When finished dumping /var, unmount and delete the mount point]
19. ___ **/sbin/umount /clone**
20. ___ **rm –rf /clone**

The resulting disk is now an exact replica of the system disk prior to being placed into service. To verify this backup, replace the original system disk with the cloned disk and restart the system. After the machine has started, run Tripwire in the integrity checking mode to verify that the disk is identical to the original. Tripwire should not report any additions or changes! Shutdown the machine, reinstall the original disk and store the clone for future use. Whenever the machine is updated in the future, this backup disk should be kept up-to-date by cloning.

Regular Full and Incremental Backups

1. ___ Include this machine in the daily tape backup program at your site.

Physical Security

Having done all to stand against the hackers, do not forget to secure the machine from unnecessary and unauthorized physical access.

1. ___ All SGI workstations are equip with a metal locking bar that extends from the front to the rear. This should be fitted with a padlock to prevent unauthorized opening of the machine to reset jumpers or to remove and install disks or CD-ROM drives. Use it!
2. ___ Machines located in public areas should be monitored for excessive reboots that may indicate an attempt to gain root access.
3. ___ Users should be reminded not to leave their workstations unattended without first logging off or using the xlock feature.
4. ___ Users should be made aware of site's security policy in writing at the time the account is issued. They should be required to express their agreement with such by signing and dating a specially prepared and authorized form. A copy should be kept in their file.
5. ___ Physically secure all backup-media in a locked safe or its equivalent. In addition, a copy of a recent full backup should be kept safely offsite in the event of some catastrophic act of God …..or man.

TESTING

The proof of the pudding is in the eating! To prove that this machine will not be just another "road-kill" on the information superhighway, it must be tested.

1. ___ If the site has a security department, have them point their ISS, CyberCop or Nessus at this host and fire away. If not, you may have to run the scanner your self. If there are remaining vulnerabilities they should be found.

At the machine, there are several tests that should be performed.

2. ___ Power cycle the machine repeatedly and try to access single-user access without supplying the PROM password. [You should not be able to]
3. ___ Try to login as root at the console or from a remote station using telnet, ftp, rsh, rlogin and ssh. [You should not be able to]
4. ___ Try to ssh into the machine as a non-privileged user. [You should succeed]
5. ___ Try to ssh out to another machine. [You should succeed]
6. ___ Try to send mail out. [You should succeed]
7. ___ Try to access the Internet. [You should succeed]
8. ___ Try to create a core file with "kill –QUIT <PID>". [You should not be able to]
9. ___ Try to write to /usr. [You should not be able to]
10. ___ Try to execute a setuid script on /var. [You should not be able to]
11. ___ Check /var/adm/authlog and /var/adm/loginlog for relevant entries. [Should be recording login info]
12. ___ Verify that no NFS, NIS or RPC services are running.
13. ___ Edit and copy one of the system configuration files and run Tripwire in the integrity-checking mode. [It should report the addition and change]
14. ___ Try to access system logs as non-privileged user. [You should not be able to]

After confirmation from the security department that the machine was found road-worthy, setup a regular testing schedule and bring the machine online.

APPENDIX A
do_not_install list

IRIX products that are selected by default, but should not be installed.

Welcome	Customer Welcome, August 2000
Register	On-Line Registration, 2.1
roboinst	RoboInst Tools for Automatic Installations 1.2
pcp_eoe	PCP EOE Software
performer_demo	Performer2.2.6 Demos and Demo Data
InPerson	InPerson Desktop Conferencing Software
appletalk	Xinet Macintosh Connectivity 10.02
demos	Demonstration Programs, 6.5
macromedia	Macromedia Movie Player, 1.4.1
netwr_client	NetWare Client 1.1
nss-fasttrack	Netscape Fasttrack Server, 3.03
outbox	OutBox Personal Web Site, 1.6
sgimeeting	SGImeeting Collaboration Environment, 1.2
sitemgr	SiteMgr - Web Content Administration, 1.1
websetup	Web Setup and Administration, 3.1.1
infosearch	Information Searching Software

APPENDIX B
Default Audited Events

sat_access_denied	Access to the file or some element of the path was denied due to enforcement of MAC or DAC permissions.
sat_domainname_set	The domain name was set.
sat_mount	A filesystem was mounted or unmounted.
sat_ae_custom	An application-defined event occurred. Application developers can engineer their applications to generate this event.
sat_exec	A new process has been introduced by exec.
sat_open	A file was opened with write permission.
sat_ae_dbedit	A file was modified using the dbedit utility. (This utility is available only with the Trusted IRIX/B optional product.)
sat_exit	The user ended the current process.
sat_proc_attr_write	The user finalized a change to a process's attributes.
sat_ae_identity	A login- or logout- related event occurred
sat_fchdir	The user changed from the current working directory to the directory "pointed" to by the given open descriptor.
sat_ae_mount	An NFS filesystem was mounted.
sat_fd_attr_write	The user changed the attributes of the file "pointed" to by the given file descriptor using fchmod.
sat_bsdipc_create	The user created a socket.
sat_file_attr_write	The attributes of a file were written by chmod.
sat_proc_read	The user read from a process's address space using ptrace.

APPENDIX B continued

sat_bsdipc_create_pair	The user created a socket pair.
sat_file_crt_del	A file was added or removed from a directory.
sat_proc_write	The user finalized a changes to a process's address space using ptrace.
sat_file_crt_del2	This is the same as sat_file_crt_del, but reports that two files (perhaps a link) were removed.
sat_svipc_change	The user set some attribute of a System V IPC data structure.
sat_bsdipc_mac_change	The user changed the MAC label on a socket.
sat_file_write	The data in a file was modified by truncate.
sat_svipc_create	The user created a System V IPC data structure.
sat_bsdipc_shutdown	The user shut down a socket.
sat_fork	The user duplicated the current process (thereby creating a new process).
sat_svipc_remove	The user removed a System V IPC data structure.
sat_chdir	Current working directory was changed with chdir.
sat_hostid_set	The host ID was set.
sat_sysacct	System accounting has been turned on or off.
sat_chroot	Current root directory was changed with chroot.
sat_hostname_set	The hostname was set.
sat_tty_setlabel	The user set the label of a port via ioctl.
sat_clock_set	The system clock was set.

APPENDIX C
Adding Files to the Volume Header With dvhtool

The volume header of system disks must contain a copy of the programs sash. The procedure in this section explains how to put sash or other programs into a volume header.

When programs are added to the volume header of a disk, there are two sources for those programs. One is the /stand directory of the system and the other is the /stand directory on an IRIX software release CD. The /stand directory on a CD (usually /CDROM/stand after the CD is mounted) contains copies of sash, fx, and ide that are processor-specific.

As superuser, perform this procedure to add programs to a volume header:

1. Invoke dvhtool with the raw device name of the volume header of the disk as an argument; for example:

```
# dvhtool /dev/rdsk/dks0d2vh
```

2. Display the volume directory portion of the volume header by using the vd (volume directory) and l (list) commands:

```
Command? (read, vd, pt, dp, write, bootfile, or quit): vd
(d FILE, a UNIX_FILE FILE, c UNIX_FILE FILE, g FILE UNIX_FILE or l)?
        l
    Current contents:
        File name       Length    Block #
        sgilabel        512       2
        sash            159232    3
```

3. For each program that you want to copy to the volume header, use the a (add) command. For example, to copy sash from the /stand directory to sash in the volume header, use this command:

```
(d FILE, a UNIX_FILE FILE, c UNIX_FILE FILE, g FILE UNIX_FILE or l)?
        a /stand/sash sash
```

As another example, to copy sash from a CD to an IP20 or IP22 system use this command:

```
(d FILE, a UNIX_FILE FILE, c UNIX_FILE FILE, g FILE UNIX_FILE or l)?
        a /CDROM/stand/sashARCS sash
```

4. Confirm your changes by listing the contents of the volume with the l (list) command:

```
(d FILE, a UNIX_FILE FILE, c UNIX_FILE FILE, g FILE UNIX_FILE or l)?
        l
    Current contents:
        File name       Length    Block #
        sgilabel        512       2
        sash            159232    3
```

5. Make the changes permanent by writing the changes to the volume header using the quit command to exit this "submenu" and the write command:

```
(d FILE, a UNIX_FILE FILE, c UNIX_FILE FILE, g FILE UNIX_FILE or l)?
        quit
Command? (read, vd, pt, dp, write, bootfile, or quit): write
```

Quit dvhtool by giving the quit command:

```
Command? (read, vd, pt, dp, write, bootfile, or quit): quit
```

REFERENCES

[1] Pomeranz, Hal (ed), "Step-by-Step Solaris Security Version 2.0", SANS Institute, 2001.

[2] Johnson, Karen (et al), "IRIX Admin: Backup, Security and Accounting (document: 007-2862-004)", Silicon Graphics Inc., 1996-1999.

[3] Pomeranz, Hal (ed), "Step-by-Step Solaris Security Version 2.0", SANS Institute, 2001.

[4] Schlagel,Thomas J., "Network Time Protocol – Version 3", CCD/BNL, 1995.

[5] Ellis, Susan & Lavine, Steven, "IRIX Admin: Disks and File Systems (document: 007-2825-007)", Silicon Graphics Inc., 1999-2000.