Global Information Assurance Certification Paper

IT Security on SGI Systems running IRIX 6.5.x

Thomas Gaeng

October, 2004

Practical Assignment for the SANS
GIAC Security Essentials Certification (GSEC)
Version 1.4c (August 2004)
Option 2 (Case Study in Information Security)

Course "Track 1c: SANS Security Essentials and the CISSP 10 Domains"
taken at SANS "Bootcamp" Meeting in Baltimore, MD in May 2004

Contents:

Abstract

This paper expands on the topic of "UNIX Security" as introduced in section 1.6 of the SANS "Security Essentials" course notes. The course notes give a broad introduction on securing UNIX systems with emphasis on LINUX and SOLARIS systems, while this paper serves as a detailed and specific guide to securing a SGI workstation running a version of the IRIX 6.5.x operating system.
In the following sections I will lay out the motivation and give a general introduction ("Before"), then describe in detail how to patch and update the IRIX operating system to the newest relevant release levels and include a list of specific steps to secure IRIX systems ("During") and will finish with a summary of the resulting state of the system and what future steps need to be taken to keep the system on a high security level ("After"). I will assume that the most basic security steps, like changing the original passwords as delivered out-of-the-box, have been carried out and that the SGI computer is connected to the internet, i.e., the necessary networking steps have been performed correctly.

"Before"

Motivation

The motivation behind the paper is that, although keeping the course material general is good for teaching and learning about IT security, in the actual workday it is often more useful to have specific guides that provide short and concise "hands-on" steps to secure specific operating systems.
The goal of this paper is to contribute an "IRIX" version of a Step-by-Step IT security guide, similar to the ones available for example at the SANS School Store, where users or inexperienced system administrators can go to find their specific operating systems and follow the steps outlined in the respective documents to improve the IT security of their operating systems.
This paper outlines the specific steps that need to be performed to improve the IT security of a computer running the IRIX 6.5.x operating system and keep it updated and patched. I chose this operating system because I have the most experience with its administration and maintenance. I have applied the steps listed in this paper on different SGI hardware workstations (e.g., Indigo, Indy, O2, Octane, …) that are connected in a heterogeneous network (SGI, PC [Windows, LINUX] and Apple [MAX-OS X] systems) with peripheral devices (e.g., printers, external disk systems, CDROM, ...) and access to the internet.
I will not explain the deviations of those steps that are necessary for operating systems older than IRIX 6.5 (SGI has stopped support for many of those older versions since the introduction and release of IRIX 6.5 in the summer of 1998).

Introduction to the IRIX operating system

IRIX 6.5 is the fifth-generation SGI UNIX operating system and is compliant with UNIX System V Release 4 and The Open Group's many standards including UNIX 95, Year 2000, and POSIX. It was released in summer 1998 as a major operating system upgrade and consolidated the support of different hardware platforms with MIPS CPUs greater than R4000 into one operating system. Before the release of IRIX 6.5 different hardware platforms were running on different operating system versions (Indy: IRIX 6.2, O2: IRIX 6.3, …) thus the new release greatly facilitated the system administration of a network with different SGI workstations! For further details see the official SGI IRIX software release web site at: http://www.sgi.com/products/software/irix/.

I strongly recommend getting a so-called "Supportfolio" account at the SGI support website at: http://support.sgi.com/ and creating a free account. This will enable the account owner to get the updates and patches as SGI releases them, sign up for email notification systems and mailing lists and also access a large amount of useful documentation including a searchable "knowledge-database".

Introduction to IRIX system security

Like every computer operating system, IRIX must make concessions to IT security for the sake of usefulness and user-friendliness (usually the more secure a system is, the less user-friendly it is), with the balance depending on the knowledge of the user and/or system administrator and the intended purpose of the respective system and its environment of use. For example a general single user workstation might be considered appropriately "secure" when configured correctly with a web and email server. When that same computer is moved into a mission-critical network with a "trust" relationship to other systems on the same subnet its server applications might be used to compromise and launch attacks to other hosts on the network. So the system's security status would have to be re-evaluated and might not be acceptable in the new environment.

I will describe the steps necessary to secure a given single SGI host system without regard to its environment as the networking issues strongly depend on the intended usage of each specific IT system.

SGI systems as purchased will arrive with the current version of IRIX which will include all previously released patches and security fixes. However "out-of-the-box" IRIX is a relatively open system that compromises typical security guidelines in order to facilitate the system setup – e.g., it has standard accounts (guest, EZsetup, …) and default passwords. These accounts can be used to set up the system (i.e., with hostname, networking information, user and group accounts, …) but then they should be disabled and passwords changed *before* the system is connected to a network.

"During"

IMPORTANT NOTES:

- Make sure that you have a working backup of the system *before* making *any* major changes to the operating system or its applications
- If possible carry out major IT system changes on a non-mission critical computer *before* rolling them out on all systems to test the behavior of the updates, patches or configuration changes
- IRIX has 2 main software tools to update and patch the operating system:
  - *Software Manager* (/usr/sbin/swmgr, a graphical tool)
  - *Inst* (/usr/sbin/inst, a text-oriented tool)

  I will use *Software Manager* to explain the system update steps and *Inst* for the descriptions of the patch installation so that both tools are covered. Similar steps are carried out by both tools – *Inst* has the advantage of text-based tools to be scriptable while *Software Manager* is more user-friendly (e.g., *Software Manager* carries out checks on disk space requirements that have to be done by hand using *Inst*). Read the **man** pages of the software installation tool of your choice to get familiar with options that are not covered in this paper.

Steps to update IRIX 6.5.x with new system releases

Keeping the operating system up-to-date is an important security step; thus it is recommended that upgrades are performed as soon as possible after their release (and successful testing on a non-mission critical system).
IRIX 6.5 upgrades are released about every 3 months in two families: the maintenance and feature releases. It is easiest to stay in the current release family when upgrading IRIX (to find out which IRIX version is installed on your system do: **uname –R**[1] on the command-line. This will show something like "6.5.25f" which stands for "IRIX version 6.5.25 feature release".). The first step to do a system upgrade is to get the new operating system version either on CD (which can be obtained by calling SGI at: 1-800-800-4SGI = 1-800-800-4744) or by downloading the newest release from the web at: http://support.sgi.com/6.5/.[2]

The CDs and web pages provide detailed information to carry out the software installation - here I will give a general guideline using the *Software Manager* GUI:

- Always read the information supplied with the respective software update and make sure your hardware configuration and disk space are sufficient for the current release before staring with the installation process

[1] Notation: I will put text that should be typed in a terminal in boldface, 11pt font and comments in parentheses and italic, 11pt font.

[2] Note that all URL references at the SGI Support websites (with http://support.sgi.com/) require a freely available "Supportfolio" account to login. URLs with http://support.sgi.com/ will not work until the login cookie is established for the current web browser session.

- Carry out the installation steps as “root” user

- Preparations before system upgrade:

  uname –R *(IRIX version information)*
  df -lk | lp *(check disk space usage)*
  hinv | lp *(check hardware and firmware information)*
  lp /etc/fstab *(check connected disk systems and mount points)*

- Disable remote user logins:

  w *(confirm that there are no users logged in)*
  echo "system down for IRIX upgrade" > /etc/nologin
  (the file /etc/nologin disables remote logins if it exists; login prints the contents of this file before disconnecting)

- Save current system configuration files:

  chkconfig > CONFIG.system *(copy current system configuration)*
  ls -al /etc > ETC.system *(copy file information of current /etc directory)*
  cp -p /unix /unix_BAK *(copy current UNIX kernel file)*

- Software-release installation process:

  ls –al /disk_tmp *(check file access on installation disk)*

  - Start and use *Software Manager* via*:*

    → Toolchest → System → Software Manager
    and enter in "Available Software field": ***/*disk_tmp**
    (this assumes that all software files that need to be installed are in the same location in the /disk_tmp directory. If not, additional directories can be added.)

    → make sure "Default Installation" is selected
    (this will activate any software marked for UPGRADE and not install any NEW software)

    → click on “Start” button

    → resolve conflicts if there are any[3]
    (document the conflict listings and resolution choices)

    → the installation process begins with using the same release family (i.e., either “feature” or “maintenance” streams)
    *(after pressing the “EXIT” button the post-installation process begins to optimize the upgraded software products, e.g., by making them quick-start ready [see “***man rqs***”, ...])*

[3] For the novice IRIX system administrator the “conflict resolution” steps and deciding which software to install are usually the most complicated decisions to make. Unfortunately there is no “quick & easy” way to describe what to do as those steps depend on the specific situation of each individual system. There is no substitute for the experience and insights of “knowing thy system”.

- BEFORE rebooting find out which configuration files have changed

  (a new /unix kernel file will be created during the reboot process. If the installation process made any undesired changes to the configuration files it is now still easy to abort the upgrade.)

  versions changed
  (list installed configuration files that have .O or .N versions)
  chkconfig > CONFIG.system2 *(copy new system configuration)*
  ls -al /etc > ETC.system2 *(copy file information of new /etc directory)*
  diff CONFIG.system2 CONFIG.system *(compare system configurations)*
  diff ETC.system2 ETC.system *(compare /etc directory listings)*

- Make any necessary corrections to the configuration files based on the results of the previous step ("BEFORE rebooting" comparisons)
- Enable remote user logins:

  rm /etc/nologin

- Reboot the system:

  Reboot

- Check the new system:

  uname –R
  ls –al /unix*
  versions changed
  chkconfig > CHKCONFIG.system3
  diff CHKCONFIG.system3 CHKCONFIG.system
  df -lk
  hinv
  cat /etc/fstab

- Document any changes made to the system

Steps to update IRIX 6.5.x with new software patches

Release updates roll all previously issued vulnerability fixes as well as system enhancements into one large installation file. While it is convenient to update and fix a large number of issues at one time, it is also risky (because configuration files that have been adapted for your systems might be changed and new features that are introduced in the release packages might conflict with your applications) and time-consuming (lots of changes need to be tested and carefully documented).
When SGI fixes a bug or vulnerability, they release patches that are usually restricted to one or a few related software packages or configuration files and fix a specific problem rather than addressing multiple issues in a large service pack. Patches and security updates don't follow a fixed release schedule but are issued whenever a conflict has been resolved. Therefore it is useful to monitor the SGI security and patch web pages at:

ftp://patches.sgi.com/support/free/security/advisories

ftp://patches.sgi.com/support/free/security/patches
http://www.sgi.com/support/security/index.html
http://support.sgi.com/colls/patches/tools/patchset/index.html

or create a "Supportfolio" user account and subscribe to the mailing list at:

http://support.sgi.com/member/tools/surfzone/subscribe.cgi

by editing the 'MyProfile' section.
After you log in to the "Supportfolio" account, the SGI support web site lets you browse patch and security release information – an especially useful feature is searching for your relevant IRIX release version at:

http://support.sgi.com/browse_request/irix_patch_browse

Again it is strongly recommended to test and install these patches on all relevant systems as soon as possible.
Patch installation instructions are very similar to the IRIX installation/upgrade steps. The release notes on the ftp and web sites or email notifications provide detailed information to carry out the respective patch installation - here I will give a general guideline using the command-line tool *Inst*:

- Always read the information supplied with the respective patch software and make sure your hardware configuration and disk space are sufficient for the current release before staring with the installation process
- Do not automatically install all patches released for a specific IRIX version; rather read the release notes to find out if the respective patch is relevant for your system or applications and only install those
- Carry out the installation steps as "root" user
- Download relevant patches to your local system's installation disk:

mkdir /disk_tmp/Patches *(create patch directory on installation disk)*
cd /disk_tmp/Patches *(change to patch directory on installation disk)*
mkdir /disk_tmp/Patches *(create patch directory on installation disk)*
wget ftp://download.sgi.com/pub/patchSG000XXXX.tar [4]
(here I use wget as an example to download relevant patch files that are archived into one tar-file)
tar xvf patchSG000XXXX.tar *(extract patch files from tar-archive)*
ls -al /disk_tmp/Patches *(list patch files)*
more README.patch.XXXX *(double-check patch release notes)*
sum –r patch* *(confirm that checksums listed in release notes and created on local system are identical)*

- Preparations before system upgrade:

uname –R *(IRIX version information)*
df -lk | lp *(check disk space usage)*
hinv | lp *(check hardware and firmware information)*
lp /etc/fstab *(check connected disk systems and mount points)*

- Disable remote user logins:

[4] Replace this URL with the filename and respective download location after identifying which patch to install; use your favorite utility to download the relevant files.

w *(confirm that there are no users logged in)*
echo "system down for patch installation" > /etc/nologin
(the file /etc/nologin disables remote logins if it exists; login prints the contents of this file before disconnecting)

- Save current system configuration files:

chkconfig > CONFIG.system *(copy current system configuration)*
ls -al /etc > ETC.system *(copy file information of current /etc directory)*
cp -p /unix /unix_BAK *(copy current UNIX kernel file)*

- Patch-release installation process:

ls –al /disk_tmp/Patches *(check file access on installation disk)*
versions | grep patch *(check installed patch files on current system and make sure there are no conflicts → READ RELEASE NOTES ! Sometimes new patches replace old ones or have conflicts with other patches or applications - in which case see footnote 3.)*
inst –f /disk_tmp/Patches
(specify the location of the software distribution that should be installed)
ls –al /disk_tmp/Patches *(check file access on installation disk)*
Inst> keep * *(keep existing software "as is")*
Inst> list *(list software in installation directory)*
Inst> inst patchSG000XXXX *(install selected patch file)*
Inst> conflicts *(resolve and document conflicts if there are any)*
Inst> go *(start patch installation)*
Inst> quit *(do post-installation steps and quit Inst)*
versions changed
(list installed configuration files that have .O or .N versions)
chkconfig > CONFIG.system2 *(copy new system configuration)*
ls -al /etc > ETC.system2 *(copy file information of new /etc directory)*
diff CONFIG.system2 CONFIG.system *(compare system configurations)*
diff ETC.system2 ETC.system *(compare /etc directory listings)*

- Some patches require a system reboot before they become active. In those cases a new /unix kernel file is usually created during the reboot process and I recommend going through the same system configuration checkup steps as listed in the previous IRIX release installation notes (i.e., see in particular the "BEFORE rebooting" section). Also examine the files that the patch is installing or changing – they are usually listed in the release notes on the web sites together with other with detailed information.
- Enable remote user logins:

rm /etc/nologin

- Reboot the system if required by installed patch:

Reboot

- Check the new system:

uname –R

```
ls –al /unix*
versions changed
chkconfig > CHKCONFIG.system3
diff CHKCONFIG.system3 CHKCONFIG.system
df -lk
hinv
cat /etc/fstab
```

- Document any changes made to the system

You'll notice that many steps are similar to the IRIX-release upgrade process. However since patch releases typically address only a specific software package the relevant files are fewer and smaller and the installation is much faster.

Steps to improve IT security on IRIX 6.5.x systems

In this section I assume that the above steps have been performed (i.e., the operating system is on the newest release level and all installed software is patched), that basic security steps like changing the original passwords as delivered out-of-the-box have been carried out and that the SGI computer is connected to the internet; i.e., the necessary networking steps are set up and have been performed correctly (the easiest way to do that is probably to use the *System Manager* GUI under the *Toolchest* as part of the IRIX *Window Manager*).

Following is a list of steps that should be checked whenever major changes to the IT systems are performed. That way configurations changes that might have occured out in the background of the installation process can be detected.

1) In **/etc/passwd** and/or **/etc/shadow**:
   - lockout unused or non-login accounts (i.e., delete or put a "*" in the password columns of the following entries: lp, EZsetup, OutOfBox, guest, 4Dgifts, ...)

   → check via:

   ```
   egrep "lp|EZsetup|OutOfBox|guest|4Dgifts" /etc/passwd
   egrep "lp|EZsetup|OutOfBox|guest|4Dgifts" /etc/shadow
   ```

2) If your system is *NOT* using NIS/yp then set up the shadow password file via:

   ```
   pwconv
   ```

   → read man pages for: **shadow, pwconv**

3) Install, set up and run *tcp-wrapper* [5]:

[5] The source code to Wietse Venema's TCP wrapper utilities can be found at various web sites – e.g.: http://ftp.porcupine.org/pub/security/ and an IRIX compiled version is available at the SGI Freeware web site under: http://freeware.sgi.com/Installable/tcp_wrappers-7.6-sgipl2.html

→ It is best to get source code, edit **Makefile** and **Banners.Makefile**[6] to enable banners and compile *tcpd* using:

make irix6 *(works fine under IRIX 6.5 system)*
make -f Banners.Makefile

→ create **/etc/hosts.allow** and **/etc/hosts.deny** files and edit **/etc/inetd.conf** to run selected daemons via *tcpd*

→ reread and restart inetd.conf via:

/etc/killall -HUP inetd

4) Make sure *ssh* services are configured correctly:

→ currently SGI has incorporated a version of openssh into IRIX

→ check via:

ssh -V
more /etc/ssh/ssh_config
more /etc/ssh/sshd_config
ps -ef | grep ssh

→ the following changes to the default version of the sshd configuration file (**sshd_config**) are more restricted while allowing general service support:

PermitRootLogin no
X11Forwarding yes
X11UseLocalhost no
Banner /etc/BANNER_MESSAGE_FILE

5) In **/etc/services**:

- comment out 2 entries for *echo* and *chargen*

→ *'echo'* and *'chargen'* services can allow a denial-of-service attack, as described, for example, in CERT advisory CA-96.01

→ check via:

egrep "echo|chargen" /etc/services

- if not using *SNMP* (Simple Network Management Protocol) also comment out snmp-ports 161 and 162 (and other related ports):

→ see e.g., SANS and CERT emails of 12 Feb 2002

→ check via:

---

[6] Banners are files that display information about the legal and proper use of a computer system or web page *before* users log in. It is usually required to have banners installed on systems in order to successfully prosecute unauthorized users who improperly use the system. In particular those banners must inform users that the system is being monitored to detect improper use and other illicit activity and that there is no expectation of privacy while using this system. More Information on setting up banners on various computer systems is available at: http://www.ciac.org/ciac/bulletins/j-043.shtml

egrep 'snmp|161|162' /etc/services

- if not using *xdmcp* (X Display Manager Control Protocol) also comment out xdmcp-port 177 (and other related ports):

→ see: http://www.procheckup.com/security_info/vuln_pr0208.html
http://www.kb.cert.org/vuls/id/634847

→ check via:
egrep 'xdmcp|177' /etc/services

6) In **/etc/inetd.conf**:
   - add -h to *telnetd*
   - comment out *telnetd* (and possibly *ftpd*) if *ssh* is running
   - add -l (log all option) to *tftpd*
   - comment out the two entries for *echo* and *chargen*
   - comment out entries for *fingerd*, *rwalld* and *rusersd*

   → check via:
   egrep "telnetd|ftpd|echo|chargen|fingerd|rwalld|rusersd" /etc/inetd.conf

   - include *tcp-wrapper* for remote services (i.e., *ftp*, *telnet*, *shell*, *login*, *exec*)

   → check via:
   egrep "tcpd " /etc/inetd.conf
   ls –l /var/adm/tcpd.log *(check tcp-wrapper log entries)*

   - more entries can usually be commented out as those services are not required but come activated by default. Here are the active entries of a typical inetd.conf file that I use:

```
sgi-dgl stream tcp nowait root/rcv /usr/etc/dgld dgld -IM -tDGLTSOCKET
mountd/1,3 stream rpc/tcp wait/lc root /usr/etc/rpc.mountd mountd
mountd/1,3 dgram rpc/udp wait/lc root /usr/etc/rpc.mountd mountd
sgi_mountd/1 stream rpc/tcp wait/lc root /usr/etc/rpc.mountd mountd
sgi_mountd/1 dgram rpc/udp wait/lc root /usr/etc/rpc.mountd mountd
sgi_fam/1-2 stream rpc/tcp wait root ?/usr/etc/fam fam
sgi_pcsd/1 dgram rpc/udp wait root ?/usr/etc/cvpcsd pcsd
sgi_pod/1 stream rpc/tcp wait root ?/usr/etc/pod pod
```

   - activate changes in /**etc/inetd.conf** via: **/etc/killall -HUP inetd**

7) Add file **/etc/ftpusers** with the following ownership and permissions:

-rw-r--r-- 1 root sys 146 Nov 6 09:26 /etc/ftpusers

→ if this file exists ftp connections from accounts that match one of its entries

will not be allowed

→ typical contents:

root, sysadm, diag, daemon, bin, uucp, sys, adm, lp, nuucp, auditor, dbadmin, rfindd, EZsetup, demos, OutOfBox, guest, 4Dgifts, nobody, noaccess, sgiweb, cflmgr

⇨ Note: put each of the above entries on one separate line

8) Change /**etc/default/login** to follow:

set CONSOLE=/dev/console
set PASSREQ=YES
set MANDPASS=YES
set SYSLOG=ALL

→ check via:
egrep "CONSOLE|PASSREQ|MANDPASS|SYSLOG" /etc/default/login

9) Disable web services if not needed:

chkconfig ns_fasttrack off
chkconfig nss_fasttrack off
chkconfig webface off
chkconfig webface_apache off
chkconfig apache off
chkconfig sgi_apache off
killall ns-httpd
killall httpd

→ killall does not always remove all processes; in that case do "**kill -9** *PID*"
(*PID: Process Identification Number*, found via: **ps –ef | more**)
→ NOTE: **ns_fasttrack** or **apache**/**sgi_apache** needs to be configured "on" for web services but not **webface**
→ NOTE: currently IRIX includes a version of Apache-1x as default web server (SGI replaced the previously provided Netscape Fasttrack in version IRIX 6.5.12)
→ check via:

chkconfig | egrep 'fasttrack|web|apache'
ps -ef | grep httpd

10) Disable *routed* daemon:

chkconfig routed off
chkconfig | grep routed

→ Note: make sure there is no file **/etc/init.d/network.local**

⇨ **/etc/init.d/network.local** is pre-IRIX 6.5 setup to handle static routes and can start *routed* even when "**chkconfig routed off**"

⇨ **/etc/init.d/network.local** functionality is now handled by file **/etc/config/static-route.options**

11) Disable *timed* daemon: (use *ntp* instead to provide time synchronization)

```
chkconfig timed off
chkconfig timeslave off
chkconfig ntp on
chkconfig | grep timed
```

→ make sure *ntp* runs instead:

```
ps -ef | egrep 'ntp|time'
```

12) Disable *array* daemon:

```
chkconfig array off
chkconfig | grep array
```

13) Disable *esp* daemon:

```
chkconfig esp off
chkconfig | grep esp
```

14) If not using PCP (SGI's Performance Co-Pilot) then disable *pcmd*:

```
chkconfig pmcd off
chkconfig | grep pmcd
```

→ Note: *pcmd* runs on port 4321 and can trick some vulnerability scanners (e.g., ISS scanner) to think *rwhosid* is running

⇨ check via: **ps -ef | grep pmcd ; fuser 4321**

15) Disable *sesdaemon*:

```
chkconfig sesdaemon off
chkconfig | grep sesdaemon
ps -ef | grep sesd
```

→ Note: *sesdaemon* is a service that is frequently re-activated after IRIX system upgrades or some major patch installations

16) If not using *LDAP,* disable service:

chkconfig ldap off
chkconfig | ldap esp

17) If not using *Teleffect,* disable service:

chkconfig tfxdoff
chkconfig | tfxdesp

18) Disable *xdmcp* broadcasting:

→ in file **/var/X11/xdm/Xaccess** comment out following lines:

#* #any host can get a login window
#* CHOOSER BROADCAST #any indirect host can get a chooser

→ and restart *xdm* via:

/etc/init.d/xdm stop
/etc/init.d/xdm start

→ check via:

grep "any host can get a login" /var/X11/xdm/*
grep "BROADCAST" /var/X11/xdm/*

→ Note: for details on the *xdmcp* vulnerability see*:*
http://www.procheckup.com/security_info/vuln_pr0208.html

19) Comment out "**xhost +**" in:

/usr/lib/X11/xdm/Xsession
/usr/lib/X11/xdm/Xsession-remote

→ make sure there is no "**xhoston**" flag file in **/usr/lib/desktop**
→ make sure there is there is a "**secure**" entry in **/var/X11/xdm/Xservers**
→ check via:

xhost
grep "xhost" /usr/lib/X11/xdm/Xsession*
ls -al /usr/lib/desktop/xho*
grep "xhost" /usr/lib/desktop
cat /var/X11/xdm/Xservers

20) If not providing email service on the system, disable *sendmail* daemon:

chkconfig sendmail off
chkconfig sendmail_cf off
chkconfig | grep mail
ps -ef | grep mail

→ do this on systems that are not mail servers - don't do it if you want email forwarding enabled or if the IRIX system should send out emails (e.g., cronjob messages)
→ if you need an email service on the system do "**chkconfig sendmail on**" and make sure you run an up-to-date and secure *sendmail* version –

either by compiling it from the source code or using the newest patched version from SGI and configure it according to your needs
→ the easiest test for open mail relays is to: **telnet mail-abuse.org**
→ for details on configuring *sendmail* I refer to: http://sendmail.org/

21) Make sure the permissions/ownership of the **/tmp** directory are **1777**:

→ sticky bit "1" prevents users from deleting each other's files
(1777 = rwxrwxrwt are the default permissions of **/tmp**
and are set in **/etc/init.d/rmtmpfiles**)
→ check via:
ls -al / | grep tmp

22) Make sure the permissions/ownership of the **/var/mail** directory are **1777**:

→ "drwxrwxr-x" are the default permissions of **/var/mail** and can be changed to "drwxrwxrwt" via": **chmod 1777 /var/mail**
→ check via:

ls -al /var/ | grep mail

23) Make sure only intended entries are in **/etc/hosts.equiv**:

→ examples for trusted entries are: local hosts on subnet
→ check via:

cat /etc/hosts.equiv

24) Make sure **"/dev/ipfilter"** shows permissions of 600 (crw-------):

→ problem might be with /dev/MAKEDEV scripts of older IRIX versions; see: ftp://patches.sgi.com/support/free/security/advisories/20020408-01-I
→ check via:

ls -l /dev/ipfilter

I keep the above list up-to-date by adding items as new security issues are known and resolved. I leave the list intentionally short and in a text-oriented format so I can easily use it to check my systems after any major software upgrade or system change. The short notes in the lists are only meant as references and reminders – detailed information should always be obtained from the source of the respective application since software development makes progress and configuration options change over time.

"After"

Summary

More restrictions can of course be set up, but the previously listed steps together with replacing unencrypted remote login services (*telnet*, *ftp*, *rsh*) with encrypted session daemons (*ssh*) are good starting points to secure a SGI/IRIX workstation and leave the system in a relative open and useful state for users to access system and network services (email, web, printing, mounting disks, …).
The steps listed in the "During" section can be automated via scripts, adapted for the specific purpose of each host system and even remotely executed. For general user workstations it might be most time efficient to create a "master" image of a system and clone it to all hosts of similar purpose, then just change the system ID and user account information to deploy major system updates. However, in my experience where I deal with relatively few specialized IT systems (~5-10 SGI computers plus non-SGI systems) where most systems have very different usages, it is most efficient to create and follow typical installation processes for each specific host. I also find it very useful to set up a general "installation" disk that I mount temporarily to the system that is to be updated. That disk contains all the IRIX updates, patches or self-compiled software for the installation and one just needs to point the installation program (i.e., *Software Manager* or *Inst*) to that location and then follow the respective steps for the software installation.

Keeping the systems secure

IT security is a dynamic field so a system administrator has to keep ever vigilant; i.e., you have to monitor the relevant IT news releases to see if there are IT security vulnerabilities that affect your systems ["automated" by subscribing to relevant newsgroups], you have to monitor the IT systems to see if you are under probes or attacks ["automated" by running host-based Intrusion Detection Systems (IDS) or home-grown programs (i.e., executable UNIX shell scripts running via "cron" or "at" jobs can check live system status and/or various logfiles and notify you via email or pager systems)]. It is also a good idea to run vulnerability scanning software after every major system change and in regular intervals (every 3 months is in my experience a good time period).
The lab exercises and notes of the "Security Essentials Cookbook (SEC)" and in particular the bootable "KNOPPIX with Security Tools" CD are excellent tools to start and learn IT security. I consider open-source IT security and UNIX administration tools as viable alternatives to commercial tools especially for smaller projects where the budget for IT security is limited. I recommend that novice system/security administrators first apply any security tools on a non-mission critical system or even a small network (which could be built using e.g., VMware as an inexpensive virtual network) to gain experience before using them on important systems as mistakes can have severe consequences – again having a full backup of the systems is essential to recover from possible system compromises.

Note that when SGI retires the support of older hardware with new operating releases it will often continue to issue IT security patches. At that point one has to evaluate the importance of keeping that particular IRIX system running and what purpose it will be used for. E.g., disconnecting the system from the internet will immediately relieve a large amount of IT security concerns so that the workstation can still be used in relative safety for a long period even after the operating system is no longer officially supported.
Another important issue is network security (firewalls, router filter, switch configurations, …). Since generally most system are connected in a network, host security is of course affected by the network security. However these issues go beyond the scope of this paper and have thus not been addressed here.

Installing applications that are not part of the operating system

Many useful open-source software projects have been compiled into IRIX compatible binary versions and are available, e.g., at the SGI-supported web site: http://freeware.sgi.com. Sometimes SGI incorporates software after they have gathered experience in the freeware project into the IRIX system release (some examples are: apache, openssh, openssl, …). SGI freeware software packages are usually somewhat out of date and are configured a specific way that, although often very useful, might not be appropriate for your specific system. In that case I recommend downloading the source code of the particular software and compiling it directly on your IRIX system. Specific MIPS compiler and professional developer environments are commercially available from SGI but many recent projects successfully use the open-source *gcc* compiler (also available at the SGI freeware site).

Note on IT systems documentation

Ideally risk assessments and detailed IT security plans including contingency plans should be carried out and created in the design phase of the IT system project. Then, as the network develops, new hardware and software systems can be incorporated to follow the guidelines of the IT documents which should also provide rules on which processes and applications are allowed or required on each particular system. In reality such documentation is often missing (especially on older, more established projects) However, I find that the time spent to create good documentation is well worth the effort as one can later look up exactly what was done to the system and reverse the installation or configuration changes to fix possible problems. Also scripting the installation process can serve as useful documentation and it will speed up any repetitive processes as well. Specific documentation is often required in government or corporate policies and will help in many organizational issues (e.g., justifying time and money spent, suggesting new purchases or configuration changes …). Finally, if the worst should happen and a system gets compromised, good documentation is invaluable in going through the process of finding the weak points, bringing the system back into a secure operation and going through the incident response process.

References

SGI related information

- SGI specific security items can be found in:
  - SGI security FAQ in "comp.sys.sgi.misc" newsgroup, which can be found at:
    http://www-viz.tamu.edu/~sgi-faq/
  - SGI security information and software patches:
    ftp://patches.sgi.com/support/free/*security*/
    http://www.sgi.com/support/security/index.html
    http://www.sgi.com/support/security/patches.html
  - SGI related newsgroups (comp.sys.sgi.*) , archived at:
    http://groups.google.com/
  - Useful SGI security information from Mr. Olson:
    http://viz.tamu.edu/pub/sgi/software/security/olson-security
- Main SGI Web site:
  http://www.sgi.com/
- SGI Freeware site:
  http://freeware.sgi.com/
- SGI Support site (requires free account registration to log in):
  http://suppport.sgi.com/
- Popular web site with lots of SGI software and information:
  http://www.nekochan.net/
- SGI related help site with mailing lists:
  http://www.sgihelp.org

Further information

- General UNIX security items can be found in:
  http://www.cert.org/
  http://www.cert.org/security-improvement/
  ftp://ciac.llnl.gov/pub/ciac/
  ftp://coast.cs.purdue.edu/pub/tools/unix/
  http://www.cerias.purdue.edu/coast/
- bugtraq mailing list, archived at:
  http://www.securityfocus.com/
- comp.security.unix newsgroup, archived at:
  http://groups.google.com/
- KNOPPIX web site::
  http://knoppix.org/
- VMware web site::
  http://www.vmware.com