Global Information Assurance Certification Paper

Operating System Security Control for the SGI IRIX Environment

GCUX – V1.9

Steve Stern

March 28, 2003

Table of Contents

1.0 Summary

IRIX is a unique flavor of UNIX that has made its mark in the high-performance computer sector, particularly in the area of visualization. Many of the special affects made for motion pictures have been done on SGI's IRIX platforms (although Linux is making its mark as of late). IRIX is also used at a lot in the research and mapping industry. However, IRIX is made to run on special hardware and it is not cheap. A new Onyx workstation can run well over $100,000.

Regardless of its use, an IRIX workstation (or server) requires that the system have certain changes made to help protect against exploits. IRIX has its share of security advisories that must be headed and SGI does not delivered new systems in a very secure configuration. This does not however mean that an IRIX system cannot be made reasonably secured. It can, and this paper outlines some of the critical areas to be secured for an IRIX visualization workstation using IRIX 6.5.19.

2.0 System Description

The system to be secured is an IRIX Onyx workstation used for producing simulation analysis for a large aerospace firm. The software used to perform simulations is Delmia Envision. The users consist of engineers that need to log in locally and have access that allows them to control the system environment as needed to perform simulations.

Requirements

In order to do simulation analysis the system needs to have in excess of 100 GB of storage space, 100 MB network capable of full-duplex operations and 2 GB of memory. The graphics should be capable of rendering 13.1 Million polygons a second. The Operating system required is IRIX 6.5.

Hardware

The system to be installed and secured has the following specifications:

CPU	4 - 600 MHZ RISC, R14000, IP35 Processors with 4MB secondary cache.
Memory	2048 Mb
Storage	2 72 GB SCSI LVD drives
Graphics	InfiniteReality2E with eight-sample anti-aliasing
Network	Fast Ethernet – 100 MB/ Full Duplex

The system is shown in the following picture.

Figure 1 - Onyx2 (from SGI)

The system meets the requirements.

3.0 Risk Analysis

The chief concern for system security is that it be protected from unauthorized access both physically or remotely. Since there might be company sensitive simulations or models on the machine it must be secured from malicious access.

Also a concern is that the machine not be taken over and used to attack other hosts or networks.

The simulation software (Delmia) has no specific needs that would lessen the security system of the system.

Services required for the operating system include the following:

- **NTP** – to keep accurate system time
- **SSH** – Secure shell for retrieving model and simulation files on other systems and to connect back to the workstation to retrieve files.
- **TCP Wrappers** – for securing network services
- **Sudo** – for granting certain permissions to users in order to kill processes, restart the system and other tasks as needed to perform their work.

4.0 Procedure

This procedure assumes the host has a VGA monitor installed. If the system has an ASCII terminal to perform this procedure, prompts and menus appear as text on the screen.

4.1 Initial System Setup

4.1.1 Command Monitor Mode

Set A PROM Password

SGI systems have the capability of setting a "PROM" password that protects the system from being compromised before the operating system has been booted. Even if a system is physically protected it should have the PROM password set.

To set the PROM password:

- Boot the computer and enter maintenance mode either by pressing the 'Esc' key or clicking on "Stop for Maintenance" at the startup screen.

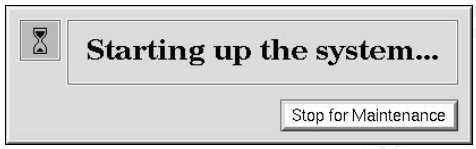

Figure 2 - Stop for Maintenance (from SGI)

- From there select the menu item "Enter Command Monitor".

Figure 3 - PROM Menu (from SGI)

- At the prompt '>> ' set a password as follows:

```
passwd
```

- Respond to the prompts for a new password:

```
enter password:
re-type password:
```

Determine the system hardware

- Determine the SCSI controller and target of **both** the CDROM and the hard disk using ***hinv***. Look for the SCSI disk and CDROM lines in the output.

```
> hinv
Disk Drive : unit 2 on SCSI controller 0
CDROM : unit 4 on SCSI controller 0
```

In this example, both SCSI devices are on controller 0, but in some cases, they are on different controllers.

- Keep track of the controller and ID of both the hard disk and the CDROM.

Partition the system using *fx*

fx is an IRIX tool for disk drive initialization and partitioning. It has two modes, normal and expert. The expert mode has more features but can be more easily misused destructively. A useful flag includes "***-l***" which tells ***fx*** to keep a log file.

- Enter ***fx*** by entering:

```
boot stand/fx  --l
```

- ***fx*** will prompt for the device name (dksc), controller, drive and LUN number. Enter the appropriate information at the prompts.

```
fx: "device-name"=(dksc)
fx: ctlr# = (0)
fx: drive# = (1) 2
fx: lun# = (0)
```

In the above example, the disk was unit 2, so 2 was entered for the drive number. Otherwise pressing Enter will accept the defaults.

- Label the drives by entering **label/create/all**
- Sync the drive by entering ***label/sync***

- Partition the system by entering ***repartition/resize***:

```
fx> repartition/resize
```

A screen will show any current partitions. For example, from a 9 GB 0^2 workstation:

```
----- partitions-----
part type     blocks              Megabytes
  0: xfs      266240 + 17815200   130 + 8552
  1: raw      4096 + 262144       2 + 128
```

```
  8: volhdr   0 + 4968           0 + 2
 10: volume   0 + 17781520       0 + 8682
capacity is 8888544 blocks
```

There will be a Warning. Continue if you are satisfied that you have taken precautions to save user data.

```
fx/repartition/resize: partition to change = (swap):
Press Enter
fx/repartition/resize: partitioning method = (megabytes...):
Press Enter
```

You should enter the power of 2 that is closest in size to your real memory size. For instance, if you have 192 MB of RAM, you can choose either 128 MB or 256 MB of swap

```
fx/repartition/resize: size in megabytes (max 2048) = (128):
```

- Enter 2048

```
fx/repartition/resize: use the new partition layout? (no):
```

- Enter yes

```
fx> repartition/resize
fx/repartition/resize: partition to change = (swap):
```

- Enter root

```
fx/repartition/resize: partitioning method = (mbytes):
```

- Press Enter

```
fx/repartition/resize: size in megabytes (max 4338) = (2372):
```

- Press Enter

```
fx/repartition/resize: use the new partition layout? (no):
```

- Enter yes

```
fx> exit
```

- Exiting ***fx*** will return you to the main menu.

4.1.2 Install the Operating System

This section explains how to load the miniroot from a distribution CD that is mounted locally on the target (host).

Installing IRIX on a system that does not have an operating system installed requires a bootable filesystem called a “miniroot”. Miniroot relies on services in the hosts programmable read-only memory (PROM) to transfer special installation tools.

To install IRIX using miniroot perform to following[1]:

- Boot the computer and enter maintenance mode either by pressing the 'Esc' key or clicking on “Stop for Maintenance " at the startup screen.
- Choose **Install System Software** from the System Maintenance menu.
- Enter the PROM password entered in the PROM Password section. Press Accept to have system take the password.
- Specify the location of the installable software. In this instance choose “Local CD-ROM”
- Insert the CD labeled “IRIX 6.5.19 Installation Tools and Overlays“. and click “OK”. The system will load files and reboot.

Using Inst

For miniroot installations, ***inst*** is automatically started when the miniroot is loaded. The following diagram shows the Inst menu.

```
Default distribution to install from: magnolia:/dist/6.5/c

For help on inst commands, type "help overview".


Inst 3.4 Main Menu

 1. from [source ...]            Specify location of software to be installed
 2. open [source ...]            Specify additional software locations
 3. close [source ...]           Close a software distribution location
 4. list [keywords] [names]      Display information about software subsystems
 5. go                           Perform software installation and removal now
 6. install [keywords] [names]   Select subsystems to be installed
 7. remove [keywords] [names]    Select subsystems to be removed
 8. keep [keywords] [names]      Do not install or remove these subsystems
 9. step [keywords] [names]      Interactive mode for install/remove/keep
10. conflicts [choice ...]       List or resolve installation conflicts
11. help [topic]                 Get help in general or on a specific word
12. view ...                     Go to the View Commands Menu
13. admin ...                    Go to the Administrative Commands Menu
14. quit                         Terminate software installation
```

Figure 4 – INST Menu (From SGI)

- Since *FX* was used to repartition the disk drive, you will see the following prompts:

Make New Filesystem on /dev/dsk/dks0d1s0? :

Enter **yes**

Are you sure? [y/n] (n):

Enter **yes**

Block size? :

Enter **4096**

- Set the system date if necessary

  Inst> admin date mmddhhmmyy

- Specify the source for the IRIX software by using the ***from*** command:

  from /CDROM/dist

This sets the source for the installation to the distribution point on the CDROM. Some notes will display on the screen, which should be read and headed since there are important steps and warnings included.

- Press the space bar to have **Inst** read the products descriptions.
- The **Inst** program will report **"Reading product descriptions .. 100% Done**." when all products have been read in successfully. Press Enter to accept the default distribution point (CDROM/dist)
- Insert CD's as necessary when requested.
- When all CD's have been read enter ***conflicts*** to see if there are any problems of dependencies that need to be addressed. There should be none if you put the correct CD's in.
- Enter ***go*** to have Inst determine whether the selections contain incompatibilities, missing prerequisites, space shortages, or other errors that might make the new software configuration unsuitable for the target.
- At the prompt "***You may continue with installations or quit now***." Look to see a line above that that says "***Installations and removals were successful***" Enter ***quit*** to have the system reconfigure the system with the new software. The screen will display a message like follows:

  Requickstarting ELF files (see rqsall(1))...100% Done.
  Automatically reconfiguring the operating system.

4.1.3 System Upgrades and Patches

The importance of keeping a machine up-to-date on the latest security fixes cannot be overstated. SGI periodically release security advisories that state the impact and the affected systems. Keeping informed of security advisories and their fixes can be done by joining the SGI Security Mailing list, known as Wiretap, or by going to the SGI Security home page located at http://www.sgi.com/support/security/

Subscribing to SGI's Wiretap mailing list can be done via the Web at http://www.sgi.com/support/security/wiretap.html

When a fix is released, either installing a patch or an overlay can update the operating system. Overlays are intermediate OS upgrades that include all applicable patches up to that point and any new ones. Patches are released to fix a specific problem before the next overlay released.

Overlays

Overlays include two types of release streams – maintenance and feature.

> **Maintenance** - includes accumulated bug fixes and basic support for new hardware and hardware upgrades while maintaining compatibility and stability. Maintenance overlays are usually released every three months.
>
> **Feature** - typically contains all of the same bug fixes and hardware support as the maintenance stream, plus new software features. It is a superset of the maintenance release stream, and it is the basis for the next major release. Feature overlays are release on a semi-annual basis.

For reliability and stability reasons it is recommended that the maintenance stream be used instead of the feature stream.

To install an overlay:

- Obtain the latest overlay from either a download on SGI's Supportfolio site http://support.sgi.com or from CD if the system is on an entitlement.
- Make the CD distribution (dist) points available.

If downloaded and uncompressed, the distribution point will be the root of the folder created from the uncompressed file. For instance, if the file downloaded is ***IRIX6.5.19_1of4.tar*** and this file is uncompressed to **/tmp/6.5.19/dist1** the dist1 is the first distribution point. The next three downloaded overlays can be uncompressed to dist2, dist3 and dist4. For CDROM simply substitute **/CDROM/dist** for **/tmp/6.5.19/dist1.** Do not ***cd*** into the CDROM or you will not be able to eject the CDROM to insert a new one. For the downloaded files it really does not matter but it is usually better to ***cd*** into a level just above the distribution points (**/tmp**).

- Use ***/usr/sbin/inst*** to load the overlays

Downloaded: **/usr/sbin/*inst –f /tmp/6.5.19/dist1***

CDROM: **/usr/sbin/*inst –f /CDROM/dist***

- At the ***inst>*** prompt, use the ***open*** command to read the overlays.

  Downloaded: ***open /tmp/6.5.19/dist1***

  CDROM: ***open /CDROM/dist***

  There should be four overlays to read. Use ***open*** to load each one until all have been opened. Eject the CDROM and insert a new CD as required.

- After all overlays have been loaded, Select the recommended upgrades from the overlay and install them.

```
inst> keep *
inst> install standard
inst> go
```

- Type ***exit*** to have the system re-read the ELF file descriptor.
- Reboot the system as requested
- Check the release and verify the OS is the latest in the maintenance stream.

```
/sbin/uname -R
```

  It should report back the base release level and extended or intermediate release level. For example if the following is run at a command prompt:

```
6.5 6.5.19m
```

  "6.5" is the base release level and "***19m"*** is the intermediate release level.

It is notable that 6.5.19 overlay contains OpenSSH version 3.4p1 but is not configured.

Patches

SGI Security Advisories contain workaround solutions or fixes to specific vulnerabilities. The fixes are often in the form patches. There are two types of patches from SGI: Required/Recommended patches; which should be installed; and Fix-on-Fail patches; which are patches that should only be installed if needed to fix a particular problem.

Patches are downloaded from SGI's Supportfolio site http://support.sgi.com. Patches should be downloaded and tested on a non-productions system before installation on production machines.

To determine which patches are installed can be done by using the following command:

```
usr/bin/versions -a | grep -i patch
```

To load a new patch read the notes associated with the patch. Most use **inst**::

- Download the patch and uncompress it if nessesary.
- Open **inst**
- At the Inst> prompt, type

  ```
  install patchSGxxxxxxx
  ```

  Replace xxxxxxx is the patch number.

- Initiate the installation.

  ```
  inst> go
  ```

- See if there are any conflicts

  ```
  inst> conflicts
  ```

- Address conflicts by either deselecting conflicts or canceling patch loading. For example, to keep a patch and deselect others:

  ```
  inst> keep patchSGxxxxxxx
  ```

- Exit **inst**

  ```
  inst> quit
  ```

- Reboot the system if requested to do so.

4.2 System Setup

The system will be first set up to a basic configuration before any of the more advanced features are added.

Assumptions

- When entering commands it is important to use the full path to the utility since a compromised version of the command may exist elsewhere in the path.
- All of the commands used in this section assume that root is logged in.
- When using the ***inst*** program, all packages are to be removed afterwards. They will not be reused. This is done by pressing "**2**" *(Remove this distribution*) to

answer the "*Do you want to save this distribution for future installations*" prompt after installing a package.

4.2.1 Basic Configuration

Set a root password

- Log in as root and immediately set a root password:

```
/usr/bin/passwd
```

Set system hostname

- Enter in the hostname for the machine. For example, to set the hostname to "victory":

```
/usr/bsd/hostname victory
```

- Add hostname to **/etc/sys_id**.

```
echo > /usr/bsd/hostname /etc/sys_id
```

.**Set the IP Address and associated settings**

- Edit **/etc/hosts** and put the IP address for the system in the first real host line with its fully qualified domain name (FQDN) first. For example, if the IP address for local host "*victory.corp.com*" is 192.168.1.1 then the **/etc/hosts** should read as follows:

```
127.0.0.1       localhost
192.168.1.1     victory.corp.com  victory
```

Note: the entry for localhost is required and should not be removed.

The hosts file should also list the local hosts fully qualified domain name and then the short name. Then list other relevant machines and their IP addresses. It is important to check for spelling errors or correct IP address numbers.

- Edit **/etc/config/static-route.options** to set the default gateway IP. For example if the gateway is at 192.168.1.254:

```
$ROUTE $QUIET add net default 192.168.1.254
```

Caution: any extra lines (carriage returns) before or after this entry could prevent networking from starting!

- Edit **/etc/config/ifconfig-1.options** to set the network mask and other settings used by ifconfig at startup.

```
netmask 0xffff0000
```

Note: this must be on a single line or networking may not work

This netmask is in hexadecimal format. For a netmask of 255.255.0.0 use the hexadecimal value 0xffff0000. For a netmask of 255.255.255.0 use 0xffffff00.

- Edit **/etc/resolv.conf** and add FQDN and at least one Domain Name Server (DNS) if DNS is to be used. For example, for domain of *corp.com* and DNS servers at IP address 192.168.44.191 and 192.168.5.191:

```
domain corp.com
nameserver 192.168.44.191
nameserver 192.168.5.191
```

One can add the following line if the domain to search DNS records for is different than the domain the machine is in. For example, to have the machine search *corp.com*:

```
search corp.com
```

Set the Timezone

- Edit **/etc/TIMEZONE** and change the relevant line entry to the correct time zone. For example, to set the machine to use the Eastern Time zone:

```
TZ=EST5EDT
```

Kernel Tunable Parameters

- Disable **ipforwarding**. Since the system will not be a router we will not enabled it because it is a security risk.

```
/usr/sbin/systune ipforwarding 0
```

- Disable the ability for users to give away ownership of files.

```
/usr/sbin/systune restricted_chown 1
```

- Disable the creation of core files.

```
/usr/sbin/systune rlimit_core_max 0
```

Core file can contain sensitive information and are really only useful for debugging purposes. Since this machine is not a developer machine we choose to disable it.

- Save the kernel parameters

```
/etc/autoconfig
```

At this point the system is configured and should be restarted in order to complete the next section.

```
/etc/reboot
```

4.2.2 User Accounts

Account Defaults

Before any accounts are added, it is necessary to configure the system to set server-wide settings that will help set user account defaults. The values that are listed here are examples and should be changed to reflect the local security policy.

Defaults for /etc/default/passwd

The configuration file **/etc/default/passwd** is used to set password length, history and aging. It does not exist by default and therefore must be created.

To make a valid **/etc/default/passwd** file

- Create **/etc/default/passwd**

```
/sbin/touch /etc/default/passwd
/sbin/chmod 644 /etc/default/passwd
```

- Add the following lines to the file:

Entry	Description
HISTORYCNT=10	Remembers password history (up to 25) by using /etc/passwd.history
HISTORYDAYS=180	Number of days to retain previous passwords (Max 730)
MAXWEEKS=12	Number of weeks before a user must change password.
MINWEEKS=1	Number of weeks before a user can change their password.
PASSLENGTH=8	Minimum number of characters in a password.
WARNWEEKS=4	Number of weeks before a password expires that the user is to be warned.

Note: the entries in **/etc/passwd** and **/etc/shadow** take precedence over the settings in /etc/default/passwd.

Default for /etc/default/su

The /etc/default/su file is used to set the defaults for users that switch users using the **/sbin/su** command.

- Create the SULOG

```
/sbin/touch /var/adm/SULOG
```

- Set permissions on SULOG

```
/sbin/chown root:sys /var/adm/SULOG
/sbin/chmod 600 /var/adm/SULOG
```

- Add/modify the following lines to the **/etc/default/su**:

Entry	Description
CONSOLE=/dev/console	Log us attempts to /dev/console
SYSLOG=all	Log all su activity
SULOG=/var/adm/SULOG	Log all su messages to /var/adm/SULOG

Default for /etc/cshrc and /etc/profile

For Bourne and Korn shell (/bin/sh) environments, when a user first logs in the file */etc/profile* is read before reading the users *.profile* file in the user's home directory. For the C shell (/bin/csh) environment these files are /etc/cshrc and .cshrc respectively. These two files set the defaults for all users but can be overridden by the respective file in the users home directory.

Add/modify the following line to the **/etc/cshrc** and **/etc/profile**:

Entry	Description
umask 022	Will cause files created by users to be rwxr-x--- or remove write permissions for group and others.

Defaults for /etc/default/login

The /etc/default/login file sets the default C shell (/bin/csh) environment login parameters.

Add/modify the following lines to the file **/etc/default/login**:

Entry	Description
SYSLOG=ALL	Logs all login attempts, not just successful ones, to SYSLOG
CONSOLE=/dev/console	Disable direct root logons over the network. Users needing remote root access will need to use su or sudo
PASSREQ=YES	Enforce that no null passwords are allowed
MANDPASS=YES	
UMASK=022	Set default UMASK so that users cannot overwrite each others files
DISABLETIME=20	Number of seconds a logon is disabled after an unsuccessful logon
MAXTRYS=3	Number of tries before a logon is considered unsuccessful
IDLEWEEKS=1	Grace period for an expired password before the account is disabled
SLEEPTIME=5	Wait (sleep) 5 seconds between failures
LOGFAILURES=3	The number of tries before an unsuccessful logon attempt is logged to **/var/adm/SYSLOG**

Default for root

Root's environment needs to be set to use a more secure umask.

Add/modify the following line to root's **cshrc** and **profile** files:

Entry	Description
umask 027	Will cause files created by users to be rwxr-x--- or remove write permissions for group and all permissions for others.

4.2.3 Passwords

Passwords are a way of authenticating users for access to the system. This method is only as secure as the password. Passwords that are easy to guess, written down or given to another person, compromise system security.

Examples of bad passwords include:

- Ones that use personal information as a password. A spouse's name, the street address, social security number or a pet's name can be easily guessed especially through social engineering.
- Ones that use common English or other foreign language words. Hackers often try these words first.
- Ones that use keyboard sequences such as QWERTY.
- Ones that are made up of a string of one letter or number, such as 333333, or a sequence of numbers, like 12345.

The use of a mixture of uppercase and lowercase letters, punctuation characters or letters and numbers mixed with keyboard symbols is highly encouraged.

Users should be trained to create a secure password scheme. Such a scheme may include a mnemonic method where instead of words a user may select a familiar phrase to create a difficult to guess password.

For example:

A trip to the moon and back

Could be used to create a password of:

a3tm%n&<

Note that IRIX systems only use the first eight characters of a password. Any extra characters are dropped and are not used as part of the password. For example, a nine character password is equivalent to only the first eight characters.

Password File

Passwords are kept in **/etc/passwd**. This file must have world-readable permissions (644).

This file uses the following format:

```
username:password:uid:gid:comments:home directory:shell
```

The **username** is the users account name they are to use to log on.

The **passwd** is to be set later so for now just put a "*" character to lock it until a password is set.

The **uid** is a number that the system actually uses to identify the user. The username is a user-friendly mapping of the uid.

A normal user id is a unique number between 100 and 65534. The following table summarizes uid numbers:

UID	Description
0	superuser
1-10	Daemons and pseudo users
11-99	System, reserved and "famous" users
100 -65534	Normal users
60001	"nobody" (occasionally 32000 or 65534)

60002	"noaccess" (occasionally 32001)

The **gid** is the group ID number the user will belong to. The group ID's are set in ***/etc/group*** and is covered later.

The **comment** field is usually the users full name.

The **home directory** field is the full path to the users initial working directory, usually **/usr/people/*username.***

The **shell** field is the program to use as Shell when the user logs in. Examples include "/bin/tcsh"

Group File

Groups are defined in the file /etc/groups. This file uses the following format:

```
group_name:password:group_id:list
```

The **group_name** field is the name of the group.

The **password** field is a placeholder for an encrypted password for the group. If null, no password is required.

The **group_id** field contains a unique numerical value for the group.

An example of an entry in **/etc/group** is as follows:

```
users:*:20
```

/etc/groups should have world-readable permissions (644).

Shadow Passwords

By default, the password file (**/etc/passwd**) is world-readable. Even though the passwords are encrypted in this file, they can be cracked offline enabling future access. The solution for this is to use shadowed passwords.

When shadowing is enabled, the encrypted password field in ***/etc/passwd*** is replaced with an "***x***"character and passwords are moved to ***/etc/shadow***, which is readable only by root.

To enable shadowing of the password file the following command is run:

```
/sbin/pwconv
```

This command should be run whenever a new user account is added.

Password File Modification

IRIX is one of the few UNIX operating systems that does not have a straightforward command line tool to add user accounts. The **/usr/bin/passwd** utility can be used to perform **/etc/passwd** file modifications but not to add users. Instead a lot of administrators rely on the **vi** utility to add users or simply use the GUI tool – *System Manager* which is on the *System* pull down.

Add User Accounts

To add a user account using **usr/bin/vi**, perform the following:

- Make a copy of the original file.

  /sbin/cp /etc/passwd /etc/passwd.backup

- Open /etc/passwd for editing

  /usr/bin/vi /etc/passwd

- Press **Shift + G** to go to the end of the file and then press "**o**" to open a new line.
- Determine the required fields including a unique **uid** and enter them in the appropriate places. For example:

  steves:*:52646:20:Steve Normal:/usr/people/steves:/bin/csh

- Create the users initial password. Enter a secure password for the user.

  /usr/bin/passwd steves

- Run password check on the password file to convert it to shadow passwords.

  /usr/sbin/pwck

- Make the users home directory

  /sbin/mkdir /usr/people/steves

- Copy the standard profile files to the users home directory

  For Bourne shell

  /sbin/cp /etc/stdprofile /usr/people/steves/.profile

For C shell

```
/sbin/cp /etc/stdcshrc /usr/people/steves/.cshrc
/sbin/cp /etc/stdlogin /usr/people/steves/.login
```

- Change permissions on the users home directory so that the user owns it

```
/sbin/chown -R steves.user /usr/people/steves
/sbin/chmod 700 /usr/people/steves
```

Caution: Do not change owners (chown) to ***userdir* /.*** otherwise the user will own the parent directory (/**usr/people**)!

- Have the user choose a new password when the login. The user would run:

```
/usr/bin/passwd
```

Removing User Accounts

Whenever a normal user leaves an organization, their account should be disable and the account recovered. Removing the accounts is the process of recovering the uid of the user and securing their files.

To remove an account:

- Lock the user account by either using //**usr/bin/passwd –l** or by editing **/etc/passwd**.
- Search for files owned by the user. For example, to search for all local files owned by user *steves*:

```
find / -local –user steves
```

This should also be done on any other servers/workstations the user might have accessed.

- Copy the files that were found to a different location or if required by a different user change the ownership of the files.
- Delete the account as required by the local security policy by editing **/etc/passwd**.

Password Resets

Password resets are part of most administrators' tasks. Root can change any user password on the system. This is done by using the **/usr/bin/passwd** command followed by the username. For example to change user *steves* password as root:

```
/usr/bin/passwd steves
```

To make a user change their password at the next logon use the ***/usr/bin/passwd*** command with the ***–f*** option.

For example, to force user ***steves*** to change his password at the next logon enter the following command:

```
/usr/bin/passwd–f steves
```

Password Aging

The system should be configured to use such functions as password aging, account lockout and reminders for users to change their passwords. Most of this was configured earlier. However there are a few command line options using the ***/usr/bin/passwd*** command that administrators should be aware of.

Note: Be aware that using some of these options override the settings used in **/etc/default/passwd.**

To set password aging use the ***/usr/bin/passwd*** command with the ***–l*** option.

```
/usr/bin/passwd -x max username
```

The value for ***max*** sets maximum number of days a password is valid before it must be changed. To set the maximum amount of time that must a user must wait before changing their command, add the ***-n*** option.

For example to make user ***steves*** change his password every 30 days and prevent him from changing it before 4 days, use the following command:

```
/usr/bin/passwd–x 14 -n 4 steves
```

Setting ***max*** value to 0 will prompt the user to change the password upon their next logon but will also remove password aging.

Accounts without Passwords

Certain accounts are shipped without a password. This is a well known exploit. These accounts should have a password set immediately unless they are to be removed in the next section:

root—Superuser

guest—Guest Account

lp—Print Spooler Owner

nuucp—Remote UUCP User

EZsetup—System Setup

demos—Demonstration User

OutOfBox—Out of Box Experience

4Dgifts—4Dgifts Account

Locking the **guest** account should also be considered unless there is a good business reason not to.

Disable Unneeded Accounts

By default IRIX installs many accounts that should be locked. There are two ways to lock an account. The first way is by using the ***/usr/bin/passwd*** command with the ***–l*** option. For example, the current entry in ***/etc/passwd*** for the account *4Dgifts* might look like this (all on one line, wrapped for readability):

```
4Dgifts: N3ZFGmq7U:3333:10:4Dgifts:/usr/people/4Dgifts:/bin/tcsh
```

The following command changes the password field of the entry in */etc/passwd* for account *4Dgifts* to `*LK*`, which blocks all logins to that account:

```
/usr/bin/passwd -l 4Dgifts
```

The password field entry now looks like this:

```
4Dgifts:*LK*:3333:10: 4Dgifts:/usr/people/4Dgifts:/bin/tcsh
```

Another way of locking an account is by editing the **/etc/passwd** file directly. Changing a password field character to any string of characters that is not used by the password encryption program to create encrypted passwords will lock an account. Examples of these characters include asterisks and semi-colons. A common practice is to put an identifying mark in the password field along with the special character. For example, to lock out an account one could use "*LOCKED* in the password field:

```
4Dgifts:*LOCKED*;:3333:10:4Dgifts:/usr/people/4Dgifts:/bin/tcsh
```

This entry will effectively lock out the account.

It is highly recommended to verify the accounts are locked out after this modification.

The following accounts should be locked if not specifically needed:

4Dgifts
adm
auditor
bin
cmwlogin
daemon
dbadmin
demos
diag
EZsetup
lp (unless the server is a print server)
nuucp
OutOfBox
rfindd
sgiweb
sys
sysadm
uucp

Tighten up root Account

It is recommended[2] to modify root's account files so that default permissions on all files created by root will be "**rw-r-----**". This can aid in protecting files that root creates. To do this edit root's .***cshrc*** and .***profile*** and set the umask to **027**.

4.2.4 Configure System Settings

Display legal notice:

It is widely believed that setting a logon warning banner can help when prosecuting unauthorized users. Edit the following files and insert an approved warning banner:

/etc/motd
/etc/issue
/etc/default/telnetd
/etc/default/ftpd

Note: Even though some of services that use these banners will be disabled later, they should still have a warning banner set in case they are restarted later a special purpose.

4.3 Network Services

As a general rule, only services that are required for business and/or operational needs should be allowed to run. Disabling services is accomplished by either editing the file ***/etc/inetd.conf*** for on-demand services and by using ***chkconfig*** for boot time services.

Inetd

The Internet Daemon (***Inetd***) is the "switchboard" for "on-demand" system services. INETD listens for connection requests and is started at boot time by ***/etc/init.d/network*** after ***/etc/inetd.conf*** is read[3].

Each line of ***/etc/inetd.conf*** takes the following form (all on one line, wrapped for readability):

```
servicename socketype protocol wait/nowait user server_program
program_parameters
```

Where:

- ***servicename*** - the service name from **/etc/services**.

- **socketype** – stream, dgram or raw.
- **protocol** - one of tcp, udp, rpc/tcp, rpc/udp or one of the others supported protocols.
- **wait/nowait** - states whether inet should wait for server program to exit before starting to listen again.
- **user** - the UID that should be used to run the server program.
- **server_program** - The full pathname of the server program.
- **program_parameters** - Parameters passed to the server program, includes arg0, i.e. the name of the program itself.

The following services will be disabled by placing "#" in front of the corresponding entry in ***/etc/inetd.conf*** if not already done:

Service	Description[4,5]
Bootp	Not needed for statically assigned IP addresses.
bootparam/1	Provides boot information to diskless clients. Not needed.
Chargen	Character generator used for testing. Could be used for a DoS attack.
Daytime	Human readable time used for testing. Could cause packet storm

	on the local Ethernet segment.
Discard	Used for testing. Could cause packet storm on the local Ethernet segment.
Echo	Used for testing. Could be used for a DoS attack.
Exec	For remotely executing commands using ***rexec***. Not desired
Finger	Returns information about a particular user account or machine. not desired
ftp	FTP is not desired
http	HTTP or web services are not desired
Login	For remote login. Will use ssh instead
mountd/1,3	RPC server which answers NFS mount requests. We will not use NFS so it is disabled.
Ntalk	For synchronous conversations between user. Not desired
rexd/1	RPC server for remotely executing commands/programs. Not desired in default configuration.
rquotad/1	For user quotas on remote machines using NFS. We will not use NFS so it is disabled.
rstatd/1-3	Kernel performance statistics. Not desired in default configuration.
rusersd/1	RPC service that returns a list of users on the network. Not desired in default configuration.
sgi-dgl	Allows a remote user on another SGI to run a GL application of a host SGI over the network. Used only for graphics library server.
sgi_espd/1	Embedded support partner. Not desired in default configuration.
sgi-fam	Allows applications such as fm, Workspace, and mailbox to keep track of changes to the file system. In the absence of fam, these applications simply poll the system. Not desired in default configuration.
sgi_mountd/1	Answers file system mount requests from remote hosts via the RPC mechanism. We will not use NFS so it is disabled.
sgi_pcsd/1	Used by CaseVision/Workshop Debugger. Not desired in default configuration.
tcpmux/sgi_printer	Used for lpsched. Not desired in default configuration.
sgi_pod/1	Printer Object Database Server. Used to provide information about remote printers via the RPC mechanism. Not desired in default configuration.
sgi_snoopd/1	Part of the NetVisualyzer network traffic data analyzer package. Not desired in default configuration.
sgi_toolkitbus/1	RPC application related to famd. Not desired in default configuration.
sgi_videod/1	Used for video devices like DIVO or DIVO_DVC boards. Not desired in default configuration.
sgi_xfsmd/1	GUI tool used to create an xfs filesystem. Not desired in default configuration.
sgi-esphttp	Enables web viewing of esp data. Not desired in default configuration.
Shell	Allows commands to be run transparently within perl. Not desired in default configuration.
sprayd/1	Records the packets sent by spray. Used for testing. Not desired in default configuration.
tcpmux	Used for non-standard services that do not have a well known ports. Not desired in default configuration.
tcpmux/sgi_scanner	Used for IMPRessario scanner. Not desired in default configuration.
telnet	Replaced by SSH
Tftp	Trivial ftp. Not desired in default configuration.
Time	Used for clock synchronization. We will use XNTP instead.

ttdbserverd/1	Tooltalk. Used by some desktop tools. Not desired in default configuration.
Uucp	UNIX to UNIX copy. Not desired in default configuration.
walld/1	Used to send messages to all logged in users. Not desired in default configuration.
wn-http	Web server. Not desired in default configuration.
ypupdated/1	used for updating NIS information. Not using NIS so not desired in default configuration.

Note: When finished editing **/etc/inetd.conf**, Inetd will need to re-read its configuration file. The following command will accomplish this:

```
/etc/killall -HUP inetd
```

Run-level Services

Services that are started (or stopped) in a particular run level are controlled by the Configuration State Checker (chkconfig). The scripts themselves are located in the ***/etc/init.d*** directory and are linked to the ***rcX*** (X = run level) scripts directory.

To disabling run level services use ***chkconfig*** to decide which services will be started at boot time.
Running **/sbin/*chkconfig*** with no arguments will list the status of each service controlled through it (on or off).

```
/sbin/chkconfig
```

Adding the service and its desired state after ***chkconfig*** will configure the service. The following command will configure ***lp*** to be off at boot time:

```
/sbin/chkconfig lp off
```

By default, run-level services will be configured as follows[6]:

appletalk	Disabled
array	Disabled
autoconfig_ipaddress	Disabled
autofs	Disabled
automount	Disabled
desktop	Enabled
esp	Enabled
fcagent	Disabled
fontserver	Disabled
gated	Disabled
ipaliases	Disabled
lockd	Enabled
lp	Enabled
mediad	Enabled
mrouted	Disabled

named	Disabled
nds	Disabled
network	Enabled
netwr_client	Disabled
Nfs	Disabled
noiconlogin	Enabled
nostickytmp	Disabled
Ns_admin	Disabled
nsd	Enabled
nss_fasttrack	Disabled
pmcd	Disabled
pmie	Disabled
privileges	Disabled
proclaim_relayagent	Disabled
proclaim_server	Disabled
proxymngr	Disabled
quickpage	Disabled
rarpd	Disabled
routed	Disabled
rsvpd	Disabled
rtmond	Enabled
rwhod	Disabled
Sar	Disabled
savecore	Enabled
sdpd	Disabled
sendmail	Disabled
sendmail_cf	Disabled
snetd	Disabled
timed	Disabled
timeslave	Disabled
ts	Disabled
verbose	Enabled
visuallogin	Enabled
vswap	Disabled
webface	Disabled
windowsystem	Enabled
xdm	Enabled
yp	Disabled
ypmaster	Disabled
ypserv	Disabled

4.4 Journaling and Monitoring

4.4.1 Log Files

Syslogd is the daemon used by IRIX to record security events to logs located in ***/var/adm***. Theses logs contain useful information that can assist in determining hardware or software problems, security violations and what are normal entries based on operations.

The location for these files in an IRIX environment is as follows:

- System events are logged by syslogd to **/var/adm/SYSLOG**. This is the default setting.
- Users using SUDO granted privileges (using the **sudo** program) are logged to **/var/adm/SUDO**. This is added functionality.
- Users using su to switch users are logged to **/var/adm/SULOG**. This is an added functionality and is configured in the Account Defaults section.
- Authorization related incidences including SSH, getty and ftp are logged to /***var/adm/AUTHLOG***. This is an added functionality

Journals (log files) should be reviewed weekly and all issues of concern researched. Obviously reviews should be done more frequently if problems are suspected

The log files must exist before they can be written to so if the defaults are changed on a system it is important to create the file first. This can be done by using the **/sbin/*touch*** command.

Syslog is a flexible in that it can be configured to:[7,8]:

- Log to a file (**/** in syslog.conf)
- Log to another host (**@** in syslog.conf)
- Mail a user (add username or usernames separated by a comma in syslog.conf)
- Send a notification to a terminal/console (***** in syslog.conf)

The file **/etc/syslog.conf** provides the defaults for logging and can be modified for more robust logging.

Syslogd collects information based on the UNIX function (called a facility) and the importance (called a level).

Facilities include:

```
user, kern, mail, daemon, auth, lpr, news, uucp, cron, local0-7,
mark,*
```

Note that **mark** will log info level messages every 20 minutes if not otherwise set in ***/etc/config/syslogd.options***.

A "*****" will select all facilities except **mark**. (example: ***.debug**).

Levels include, in order of importance:

```
emerg, alert, crit, err, warning, notice, info, debug, none
```

The format to use to log incidences is ***facility.level*** <tab>***filter*** <tab> ***action***. The use of a tab character (<tab>) between entries is mandatory since the syslogd daemon will ignore spaces (as well as "#").

The ***filter*** entry can be a path to a filter file used to select text to send to the action from a facility. The use of the ***filter*** entry is optional.

The ***action*** entry can be send messages to a log file, send to a user or send to another host.

For example:

To log ***user*** facility messages at the ***alert*** level and higher (***alert*** and ***emerg***) use the following:

```
user.alert      /var/adm/SYSLOG
```

Note that this file is limited to 49 lines in the IRIX 6.5 environment.

To Improve Logging:

- Add the following entries to **/etc/syslog.conf**

Note: when editing this file take care to add tabs <tab> where noted. Do not use spaces.

```
mark.debug  <tab> /var/adm/SYSLOG
mark.debug  <tab> /var/adm/AUTHLOG
mark.debug  <tab> /var/adm/SUDO
```

This will put a mark in the SYSLOG every 20 minutes to help notice gaps in logging. The log entry will simply be the time and a "**-- MARK –**" entry.

- Edit the entry "*debug;kern.none" to add more logging facilities.

```
*.emerg;*.alert;*.crit;*.err;*.warning;*.notice;*.info;*.debug;kern.
none  /var/adm/SYSLOG
```

This entry is to be on one line and collects most messages relating to the system in ***/var/adm/SYSLOG***. Note that the kernel messages go through a filter before going to SYSLOG because of the default setting in **/etc/syslog.conf.** This default entry should not be modified and is as follows:

```
kern.debug      |/usr/sbin/klogpp      /var/adm/SYSLOG
```

- Add an entry to collect SSH, xdm, tcpd and getty information in a separate log located at **/var/adm/AUTHLOG**.

```
# Enable logging for xdm, ssh, getty ,ftpd and rshd.
auth.debug  <tab>     /var/adm/AUTHLOG
```

This will send all authorization messages of debug and higher to this log file.

- Enable logging for SUDO to **/var/adm/SUDO** (SUDO is to be installed later)

```
local2.debug    /var/adm/SUDO
```

- Create the two new log files

```
/sbin/touch /var/adm/AUTHLOG
/sbin/touch /var/adm/SUDO
```

- Change the owner of the log file

```
/sbin/chown root:sys /var/adm/AUTHLOG
/sbin/chown root:sys /var/adm/SUDO
```

- Change permissions on the log file so that it is readable only by root

```
/sbin/chmod 600 /var/adm/AUTHLOG
/sbin/chmod 600 /var/adm/SUDO
```

Note that any changes made to ***/etc/syslog.conf*** require that syslogd be restarted.

```
killall -HUP /usr/etc/syslogd
```

Rotating the Logs

It is critical to keep log files so that they can be reviewed if needed for an investigation of a security breach, hardware/software failure or other historical analysis. Therefore log files must be rotated and archived depending on the appropriate security policy.

The default IRIX installation will rotate the **sulog** and **SYSLOG** after it reaches a certain size. The **SYSLOG** will be renamed to **oSYSLOG** and the sulog renamed to **OLDsulog**. When the next log rotation takes place the previous **oSYSLOG** and **OLDsulog** will be overwritten. This is not desirable since the previous logs are now lost. It can be fixed by creating a log rotation script and editing the root's crontab for the log rotation.

Perform the following:

- Create **logrotate.sh**

```
/sbin/touch /var/adm/logrotate.sh
```

- Edit **logrotate.sh** and add the following[9]:

```
#! /bin/sh
#
# This script rolls over all the logs specified on a weekly
# basis, and also compresses them if they are quite
# large, using gzip. Any log greater than 8 weeks old will
# be deleted. Logs should be moved copied to a remote host
# before that time.
#
#This is run by an entry in /var/spool/cron/crontabs/root
#
logdir=/var/adm
if [ ! -d $logdir ] ; then exit ; fi
gz=/usr/sbin/gzip
umask 077
cd $logdir
for log in SYSLOG auth.log mail.log local0.log ; do
        if [ -f $log ] && [ "`/sbin/stat -qs $log`" -ge 10240 ]
                then
             [ -f $log.6.gz ] && /sbin/mv $log.6.gz $log.7.gz
             [ -f $log.6 ] && /sbin/mv $log.6 $log.7
             [ -f $log.5.gz ] && /sbin/mv $log.5.gz $log.6.gz
             [ -f $log.5 ] && /sbin/mv $log.5 $log.6
             [ -f $log.4.gz ] && /sbin/mv $log.4.gz $log.5.gz
             [ -f $log.4 ] && /sbin/mv $log.4 $log.5
             [ -f $log.3.gz ] && /sbin/mv $log.3.gz $log.4.gz
             [ -f $log.3 ] && /sbin/mv $log.3 $log.4
             [ -f $log.2.gz ] && /sbin/mv $log.2.gz $log.3.gz
             [ -f $log.2 ] && /sbin/mv $log.2 $log.3
             [ -f $log.1.gz ] && /sbin/mv $log.1.gz $log.2.gz
             [ -f $log.1 ] && /sbin/mv $log.1 $log.2
             /sbin/mv $log $log.1 ; touch $log
               if [ "`/sbin/stat -qs $log.1`" -ge 1048576 ]
                     then
                          [ -x $gz ] && $gz -v $log.1
              fi
        fi
find . -local -type f -name "${log}.*" -mtime +50 -exec /sbin/rm -
rf {} \;
done
killall 1 syslogd
```

- Make the file executable:

```
/sbin/chmod u+rwx /var/adm/logrotate.sh
```

- Edit root's crontab (/var/spool/cron/crontabs/root)and add (All on one line, wrapped for readability):

```
1 1 * * 0 if test -x /var/adm/logrotate.sh; then
/var/adm/logroll.sh; fi
```

The system logs will now be retained and can be copied to a remote host or backed up to tape for archiving.

Check permissions and owner on files installed in this section

- Change permissions on the file so that it is owned andreadable only by root

```
/sbin/chown root:sys /var/adm/logrotate.sh
/sbin/chmod 600 /var/adm/logrotate.sh
```

4.5 Additional Software

4.5.1 Network Time Protocol - NTP

Having accurate time synchronization on all networked systems is critical for many reasons including log analysis, cron jobs and user accounts expirations. The Network Time Protocol, or NTP, is used for synchronizing the time between computers to within millisecond of accuracy. Generally it is recommended that an organization have just one server contact a trusted external time source and then have internal systems synchronize to it. A list of public NTP servers can be found at http://www.eecis.udel.edu/~mills/ntp/servers.html. It is highly recommended that the administrator for the remote NTP server be contacted to request permission first before using their server.

There is some confusion as to which version of NTP to use, i.e. XNTP or NTP. The FAQ at www.ntp.org states that the term XNTP refers to version 2 or 3 while NTP indicated version 4.[10] Therefore beginning with NTP version 4, the naming conventions are more straightforward, and the name of the NTP distribution itself and the names of all the programs included start with ***ntp*** (i.e. ntpd, ntpq).

To install and configure a standalone NTP client:

- Download NTPv4 (**fw_ntp-4.1.0.tardist**) from SGI's Freeware site
- Install the NTPv4 program.

```
/usr/sbin/inst -f fw_ntp-4_1_0.tardist
```

NTP Startup Script

- Edit **/etc/init.d/ntp**

```
# Add a list of NTP servers to set the boot time to.
NTPSERVS=hostname
# Location of the drift file
driftfile /etc/ntp.drift
# internal fallback address (127.127.1.0)
fudge 127.127.1.0 stratum 10
```

Note that the entry NTPSERVS can include one server or a group of servers. The syntax for multiple servers is as follows:

```
NTPSERVS="hostname1 hostname2"
```

The NTP daemon computes the drift of the system clock as compared to the server reference time. The daemon can save the drift rate to a file to have it available at the next restart.

Additionally, there should be an entry for the local clock, which can be used as a fallback resource if no other time source is available. Since the local clock is not very accurate, it should be fudged to a low stratum.

NTP Configuration Script

There should be a

- Edit the ***/etc/ntp.conf*** file and add security settings.

```
# set default flags – override for access.
restrict default notrust nomodify
# Trust 10.10.10.20 for time but no changes.
restrict 10.10.10.20 mask 255.255.0.0 nomodify
# don't trust or allow changes from other than 192.168 subnet
restrict  192.168.0.0  mask  255.255.255.0
```

These lines allow access to/from subnet 192.168.0.0 but restrict access from all other hosts.

- Check permissions and owner on files installed in this section

```
ls –al /etc/init.d/ntp
ls –al /etc/ntp.conf
```

If these files are not owned by root or permissions anything other than "**-rwx------**" then change them.

```
/sbin/chown root:sys /etc/init.d/ntp
/sbin/chmod 700 /etc/init.d/ntp

/sbin/chown root:sys /etc/ntp.conf
/sbin/chmod 700 /etc/ntp.conf
```

Checking NTP Status

The command line utility **ntpq** can be used to check the status of a NTP daemon on either the local machine or on a remote host.

For local machine:

```
/usr/freeware/bin/ntpq –p
```

For remote machine:

```
/usr/freeware/bin/ntpq -p <ntpserver>
```

This command will show a table with the status of each server defined in /***etc/ntp.conf***.

4.5.2 Sudo

Sudo (superuser do) allows system administrators to give certain users (or groups of users) the ability to run some (or all) commands as root or another user while logging the commands and arguments.[11]

Sudo is available from http://www.courtesan.com/sudo/dist or from SGI (freeware.sgi.com) in tardist format.

There are a few security considerations to keep in mind when installing sudo[12]:

- When sudo is used in a networked environment the passwords are sent across the network in plain text. To eliminate this *sudo* should be used with SSH[13] or Kerberos.
- Users with *sudo* access to a given command could use that access to change their privileges. If a user was given *sudo* access to the vi editor then the user could edit the */usr/freeware/etc/sudoers* file and assume root privileges.
- If a user's password is cracked then the cracker could assume the privileges of the user as listed in *sudoers* file.

We will not allow network sudo access, only local access. However, we want users (engineers) to be able to mount, kill any process and reboot the workstation.

Install sudo

To install the tardist format:

- Download the tardist from *freeware.sgi.com*
- Install sudo

```
/usr/sbin/inst -f fw_sudo-1.6.6.tardist
```

- Edit **/etc/syslog.conf** to have sudo to log activity. This was previously done and is covered in the section on above on logging.

Configure sudo

The ***/usr/freeware/etc/sudoers*** file is a ASCII file that contains the configuration and specifications that will define access privileges. User specifications are in the following format

```
WHO    WHERE = WHAT
```

- WHO specifies the userid or userids being granted privileges.
- WHERE defines from which machine or machines the privileges are available.
- WHAT defines which command or commands can be run.

The following syntaxes must be adhered to:

- Each user specification entry must be on its own line.
- Users can be specified by group as well as userid. The group name is started with a '%' to distinguish it from a userid.
- The full path to a command must be specified

Editing the ***sudoers*** file is accomplished using the ***visudo*** command rather than opening the file with the standard vi editor.

- To edit sudoers file

```
/usr/freeware/bin/visudo
```

- Edit to meet access requirements

```
# sudoers file.
#
# This file MUST be edited with the 'visudo' command as root.
#
# See the sudoers man page for the details on how to write a
# sudoers file.
#

# User alias specification
User_Alias ENGINEERS=smith, jones, ruffus

# Cmnd alias specification
Cmnd_Alias     SIMS=/usr/bin/kill,\
               /etc/reboot, /sbin/mount, /sbin/umount \
               /sbin/shutdown -h now

# Defaults specification
Defaults       syslog=auth
Defaults       log_year, logfile=/var/adm/SUDO
# don't run sendmail
#!mailerpath

# User privilege specification
root    ALL=(ALL) ALL
ENGINEERS ALL=SIMS

# COMMANDS For USERS
#
# Allow users to mount their CDROMS
ALL  ALL=/sbin/mount /cdrom,/sbin/umount /cdrom
```

This example gives engineers (defined as smith, jones, ruffus) permissions to kill processes, reboot or shutdown the system and to mount and un-mount all drives. It also allows any user to mount or un-mount the CDROM.

Notice that in the Cmnd_Alias section a "\" is used to continue an entry that is too long on the next line.

- Create a symbolic link from /**usr/etc/sudo** to the sudo binary for the users

```
ln -s /usr/freeware/bin/sudo sudo
```

- Check permissions and owner on files installed in this section

```
ls -al /usr/freeware/bin/sudo
```

The **/usr/freeware/bin/sudo** program should be owned by root and permissions set to "**---s--x—x**". The "s" means that sudo is a setuid program. This is because in order to run applications specified by the sudoers file the user must effectively become root for that command.

```
ls -al /usr/freeware/etc/sudoers
```

The **/usr/freeware/etc/sudoers** file should be owned by root and permissions of "***-r--r--r--***". The visudo program sets the permissions on this file when it is edited which is another good reason not to use vi to edit it.

4.5.3 SSH

As stated earlier IRIX 6.5.19 comes with SSH 3.4p1 included. It is installed in ***/usr/freeware/bin***.

A newer version of SSH can also be downloaded in tardist format from SGI's Freeware site. The latest version as of March 2003 is 3.5p1. This version places the configuration files in /**etc/ssh** although the actual ssh executable is still installed in ***/usr/freeware/bin***.

The prerequisites to installing SSH are Perl5, SSL and Zlib. Perl5 should be installed by default, if not then perl-5.6.1 can be installed from SGI's Freeware site.

To Install SSH

- Download the latest stable version of OpenSSL, Zlib and SSH from SGI's Freeware site.

  Files (current versions)

  - fw_openssl-0.9.6g.tardist
  - fw_libz-1.1.4.tardist

- fw_openssh-3.5p1.tardist
- Use ***inst*** to install all required programs.

```
/usr/sbin/inst -f /tmp/fw_openssl-0.9.6g.tardist
go
open /tmp/fw_libz-1.1.4.tardist
go
open -f /tmp/fw_openssh-3.5p1.tardist
go
done
quit
```

Create an unprivileged SSHD user account. It should not be able to login and should have an empty home directory.

- Edit /***etc/passwd*** and add sshd user.

```
sshd:*:41245:302:SSH Daemon:/var/empty:/bin/false
```

SSHD account created with a password of **'*'**, shell of **/bin/false** and home directory of **/var/empty**

- Create a SSHD group and put the SSHD user in it.
- Edit **/etc/ssh/sshd_config** and make the following changes (Uncomment the entry as necessary):

  Set entry "**ListenAddress**" to the IP address of the machine.

  Set entry "**LoginGraceTime 600**" so that the server will disconnect after this time if the user has not logged in.

  Set entry **PermitRootLogin** to no. This will prevent direct root logins and make administrators login first as an unprivileged user and then **su** to root.

  Set entry "**RhostsAuthentication** no" and "**IgnoreRhosts yes**" so that the local rhost files are not used for authentication.

  Set entry "**PermitEmptyPasswords no**" to prevent users without a password from connecting.

  Set entry "**StrictModes yes**" to have SSH check user's permissions in their home directory and rhosts files before accepting a login because their may exist world-writable files in those locations.

  Set entry "**PrintMotd**" and set it to yes. This will cause /**etc/motd** to be printed when a user logs in.

Likewise set entry "**Banner**" to a valid banner path such as **/etc/issue** or **/etc/ssh/ssh_ban**

- Check permissions and owner on files installed in this section.

```
ls –al /usr/freeware/bin
```

If not owned by root and/or permissions are not "***-rwxr-xr-x***" then do:

```
/sbin/chown root:sys /usr/freeware/bin
/sbin/chmod 744 /usr/freeware/bin
```

4.5.4 TCP Wrappers

Network services that must be enabled should be configured to use TCP Wrappers to improve security. TCP Wrappers (tcpd) monitors network requests and instead of running the service it logs the request and does certain checks before running the requested service.

TCP Wrappers can be downloaded from SGI's Freeware site or from ftp://ftp.porcupine.org/pub/security/index.html. The current version as of March 2003 is 7.6.

To install TCP Wrappers:

- Download the tardist file from *freeware.sgi.com*.
- Use **/usr/sbin/*inst*** to install.

```
/usr/sbin/inst –f /tmp/fw_tcp_wrappers-7.6-sgipl2.tardist
go
```

The TCP Wrapper executable (tcpd) is now installed in **/usr/freeware/bin/** and the configuration is kept in the files ***/etc/hosts.allow*** and ***/etc/hosts.allow***. These two files allow or deny connections based on their source address.

Configuration

- Edit ***/etc/inetd.conf*** and configure services to use tcpd instead of their normal service. Replace the normal service with /usr/etc/tcpd.

  For example to configure ftp:

  Was:

```
ftp    stream  tcp     nowait  root    /usr/etc/ftp     ftpd -l
```

  After editing:

```
ftp    stream  tcp    nowait  root   /usr/etc/tcpd     ftpd -l
```

Note: do not edit rpc-based services (rpc/tcp or rpc/udp)

- Edit ***/etc/hosts.deny*** and add blanket deny statement.

```
ALL: ALL
```

- Edit ***/etc/hosts.allow*** and add authorized connection entries.

```
ALL: 192.168.1.0/255.255.255.0
```

This entry will allow connections from the 192.168.1.0 subnet only. This can be fine tuned to allow connections based on host or service using the format:

```
service1 service2 : client1 client2
```

Where service1 and service 2 are the service names and client1 and client 2 are IP, host names or wildcards such as ***.corp.com***

- Check permissions and owner on files installed in this section.

```
ls –al /etc/hosts.*
```

If not owned by root and/or permissions are not "***-rw-------***" then do:

```
/sbin/chmod 600 /etc/hosts.allow
/sbin/chown root:sys /etc/hosts.allow
/sbin/chmod 600 /etc/hosts.deny
/sbin/chown root:sys /etc/hosts.deny
```

4.6 Maintenance

The system is to be incrementally backed up nightly, with full backups done on the last weekend of the month. The software used is Legato Networker.

The backup software is run from a central server but some operations can be performed from the local host. Users can restore files that were backed up provided they are within the "browse" period and that they were the owner of the file when it was backed up.

- To restore a file to the local workstation:

  As the user that owns the file, or root who can restore any local file run:

```
nwrecover
```

  This will bring up the Networker GUI in which a user can browse for files by date and initiate a restore, provided the tape in the tape library on the backup server.

System logs are to be reviewed daily on the workstation both to held determine if a system break in process or has occurred and to explain to the users any problems they may have noticed on the machine the previous day. Administrators can use SSH to log on over the network or by logging in locally.

To view the logs use the ***tail*** command. The ***cat*** command used in conjunction with the ***grep*** command is also very usefull.

- To view the last ten entries in a log file:

  ```
  tail /var/adm/SYSLOG
  ```

  Example output:

  ```
  Mar 28 11:36:40 6>: victory -- MARK --
  Mar 28 11:37:06 6E: victory sshd[1870]: Connection closed by
  192.168.1.100
  ```

- To increase the amount of data viewed add the number of lines after the command:

  ```
  tail -100 /var/adm/SYSLOG
  ```

  Example output:

  ```
  Mar 28 08:57:04 5D:victory ntpd[507]: ntpd 4.1.0 Mon Aug  6 01:12:08
  PDT 2001 (1)
  Mar 28 08:57:04 6D: victory ntpd[507]: precision = 15 usec
  ----------------------96 lines deleted-----------------------
  Mar 28 09:36:40 6>: victory -- MARK --
  ```

4.7 Verification

In this section we will verify some of the settings that were previously set.

Accounts

No one should be able to log in a guest because it was previously locked. Note: In this test the account has a password set, and then the account is relocked before attempting the test.

```
login: guest
Password: *******
Sorry
```

The guest account is not accessible. This is as expected.

User passwords must be at least 8 characters.

User *steves* tries to set a password that is only six characters.

```
Changing password for steves
Old password: ********
New password: ******
Password is too short - must be at least 8 characters
```

User passwords are forced to be at least eight characters. This is as expected.

Sudo

User *steves* needs to kill a process started by another user that has gone home for the day.

The process *showcase* was left open and is using resources that are needed by a simulation.

```
ps -ef | grep showcase
```

Reports:

```
joeuser  4278    1  0 07:41:32 pts/2   0:01 showcase
```

When user *steves* tries to kill the process directly it fails. Users should not be able to perform other functions not defined by ***sudoers*** .

```
kill 4278
```

Reports:

```
4278: Operation not permitted
```

User should not be able to kill other users processes. This is a built in protection of the IRIX environment.

So user *steves* reverts to the ***sudo*** program that the administrator set up.

```
sudo kill 4278

We trust you have received the usual lecture from the local
System
Administrator. It usually boils down to these two things:

        #1) Respect the privacy of others.
        #2) Think before you type.

Password: *******
```

Now when we run a process list we se that the *showcase* application is killed.

```
ps -ef | grep showcase
```

Reports nothing back since the process is killed.

Sudo allows user to perform tasks as defined in the sudoers file as expected.

Sudo logged these actions to /var/adm/SUDO:

```
Mar 28 08:02:25 2003 : steves : TTY=ttyq0 ;
PWD=/usr/people/steves ; USER=root ; COMMAND=/usr/bin/kill
4278
```

Sudo logging works as expected.

SSH

Users should be able to use SSH to connect to other servers running SSH. Users should also be able to connect back to the workstation using SSH but no one should be able to connect back as root.

User *steves* connects from another workstation to the *victory* workstation.:

```
ssh victory
Last login: Fri Mar 28 08:58:48 2003 from 192.168.100.1
```

Successful connection as expected.

However, when a user that does not have an account tries to log in:

(user bonehead is not an authorized user)

```
ssh –l bonehead victory
login as: bonehead
Access denied
```

Unsuccessful connection for unauthorized user as expected.

Users should not be able to connect as root.

User tries to connect from another workstation called *freedom* to workstation *victory*:

```
ssh victory
root@freedom's password:
```

```
Connection to victory closed by remote host.
Connection to victory closed.
```

Users cannot login directly as root through a SSH session. SSH performed as expected in this regard.

Telnet Service

Users should not be able to **telnet** to the workstation.

```
telnet victory

Connecting To victory...Could not open a connection to host on
port 23 : Connect failed
```

Therefore telnet is disabled as expected.

4.8 Conclusion

An IRIX workstation can be made reasonably secure. It requires that an administrator be diligent on patches, documenting steps taken and understanding the impact any changes may have. Typical users on these systems are more computer savvy than a PC user so they must also be watched so one does not circumvent the organizations security policy.

5.0 References

---

[1] Silicon Graphics. IRIX Admin: Software Installation and Licensing. Document number: 007-1364-130. URL: http://techpubs.sgi.com/library/tpl/cgi-bin/getdoc.cgi/0650/bks/SGI_Admin/books/IA_InstLicns/sgi_html/ch03.html. (March 25 2003)

[2] Samuel, Gary. Checklist for Installing a Secure IRIX 6.5 Workstation; 5 April 2001 URL: http://www.sans.org/rr/unix/sec_irix65.php. (March 25 2003)

[3] Silicon Graphics. Inetd Man Page (March 25 2003)

[4] Fermi National Accelerator Laboratory. IRIX Services. URL: http://www-oss.fnal.gov/policy/services.html. (March 25 2003)

[5] Plessis, Wimpie du. Internet Daemon (inetd) – What it is and Securing it. 23 April 2001. URL: http://www.sans.org/rr/unix/inetd.php (March 25 2003)

[6] Haprian, John C. Securing IRIX 6.5. August 20, 2001 URL: http://www.sans.org/rr/unix/IRIX_65.php. (March 25 2003)

[7] Stanford University Security Department. Syslog Configuration. 26 September 2000. URL: http://www.stanford.edu/group/itss-ccs/security/unix/syslog.html. (March 25 2003)

[8] Silicon Graphics. syslogd man page. (March 25 2003)

[9] McCormick, Martin. Security Configuration and Policy. 18 March 1999. http://www.sas.upenn.edu/chem/facilities/computer/sysadmin/configuration_security_policy.html. (March 25 2003)

[10] Windl, Ulrich & Dalton, David. What is NTP? 18 January 2003. URL: http://www.ntp.org/ntpfaq/NTP-s-def.htm#AEN1355 (March 25 2003)

[11] Miller, Todd. SUDO Main page. URL: http://www.courtesan.com/sudo/sudo.html. (March 25 2003)

[12] About.com. Configuring Sudo. URL: http://unix.about.com/library/weekly/aa102500c.htm (March 25 2003)

[13] Forbes, Liam . Sudo and SSH: A Scheme for Controlling Administrator Privileges and System Account Access. 11 June 2001. URL: http://www.sans.org/rr/authentic/sudo.php. (March 25 2003)