Global Information Assurance Certification Paper

SECURING IRIX 6.5.13

GSEC
Susan Gilmore
June 5, 2002

1. INTRODUCTION

This practical will explain how to secure an IRIX Server; some refer to this as hardening. First let me back up a little. My experience has primarily been with SUNOS/Solaris Operating Systems. From the everyday system administration tasks, adding user accounts to upper level designing the system architecture. Recently, my focus turned more toward system security. Nowadays, everyone needs some level of computer security, right? I started by doing what I knew best, securing our Unix server environment. Our environment is composed of Ultra 60's, 80's to higher end Enterprise Servers. This wasn't as easy of a task as I thought it would be. I learned quite a bit more about the system files, why they were used and the do's and don'ts of security. What services to use and what to turn off, what system vulnerabilities were, how hacker's think. It was quite a challenge. I suggest checking out the "Hacking Exposed, Network Security Secrets and Solutions" Second Edition book [1] (Scambray, McClure, Kurtz), for some valuable information.

Just a few weeks ago, I was tasked with hardening a SGI (Silicon Graphics) Irix box. I thought sure, it can't be much different then the Solaris boxes. So, I accepted the task. Of course, I had to start with a little bit of research. I wanted to find out the differences between the Solaris Operating System and the Irix Operating System. To my surprise, there were quite a bit of differences. And unfortunately, it was a little harder to find information on IRIX then it was on Solaris. This is why I choose my topic, to help provide others with the information they may one day be requested by their boss to research.

Let me start by explaining my definition of "hardening". This was a term that was new to me as well. Over the past six months this is what I've determined it to be, ensuring that your server is secure enough for hacker's not to intrude, but open enough for you to do your job. As Network Magazine says, "OS hardening is the black art of ensuring that all known OS vulnerabilities are plugged, and that the OS is monitored continuously."[2] "Building Blocks for OS Hardening." This white paper will give you a better understanding of what IRIX 6.5.13 entails and what is needed to ensure your server is properly hardened. Remember these are strong suggestions for hardening an IRIX box, but these suggestions are not limited to this document. For procedures on loading the Irix OS, refer to Appendix A.

2.0 SECURING IRIX 6.5.13

Start by creating your workstation in a safe environment, for instance no network connections should be used until the machine has been hardened. After loading the Operating System you then need to apply patches. Patches are provided by the manufacturer of the operating system to fix bad code, update code, and/or fix security vulnerabilities. The

vulnerabilities can be found at the following sites: http://www.cert.org/[3], http://www.incidents.org/[4], and the manufacturer's website. The security patches and the recommended patch sets should be loaded at least monthly. This is recommended by the operating system manufacturer to ensure a stable and properly functional OS release. Patches for IRIX are available on the web at: http://www.sgi.com/support/security/patches.html[5]

Or to lookup a particular patch, you can go to: http://www.support.sgi.com/colls/patches/tools/browse[6]

Once you've loaded the patches, you can begin hardening your operating system. When SGI setup the default IRIX configuration environments, they did not have security in mind. Most configurations were chosen mainly for convenience. This is where your job comes into play.

2.1 ACCOUNTS

2.1.1 REMOVING UNNCESSARY SYSTEM ACCOUNTS

The IRIX OS comes with several accounts that can be removed since they are installed without a password and they are not "locked" by default. These accounts are: cmwlogin, guest, 4Dgifts, Ezsetup, demos, and OutOfBox. When removing these accounts, remember there may be files or directories on the server with their ownership. Files and directories left without an owner ID can be exploited if not assigned to a valid account. Therefore, you will need to make "root" the owner of any files/directories that were owned by these accounts. A simple 'for' loop can be written to take care of this:

```
for CHGPERM in `egrep "cmwlogin|guest|4Dgifts|EZsetup|demos|OutOfBox"
/etc/shadow|cut –d":" –f1`
do
    find / -user $CHGPERM -exec chown root {} \;
done
```

To delete the accounts mentioned above, here is another simple 'for' loop:

```
for ACCOUNTS in `egrep "cmwlogin|guest|4Dgifts|EZsetup|demos|OutOfBox"
/etc/shadow| cut -d : -f1`
do
   /usr/sysadm/privbin/deleteUserAccount -l $ACCOUNTS
done
```

2.1.2 LOCKING ACCOUNTS

Other accounts have no password assigned to them, and should be locked down. These accounts are, but are not limited to: *lp, bin, sys, adm, dbadmin, sysadm, uucp, nuucp, sgiweb, daemon, rfindd, nobody, noaccess*. Locking the account merely prevents this "user" from logging in via telnet, ftp, or console. This account can still be accessed by a process or system call. Other accounts that need to stay active, but have no password assigned, should

be forced to set a password. If an account is locked, you will see the initials 'LK' within the account. 'NP' indicates No Password is set.
"If a login is not used or needed, disable (lock) the login. You should not remove the account, though, because of the danger of reusing the UID in the future. User ID numbers are meant to be permanently associated with the person who used the account. If you reuse the UID number, the new user may find files that belonged to the previous owner of the ID number. These files may contain "trojan horse" programs that could damage your system. You may remove the user's home directory and files (after making a backup), but you should never remove an entry from your /etc/passwd file.[7] "System Login and Account Administration / Locking Unused Logins"

See the following CERT advisory, http://www.cert.org/summaries/CS-98.06.html[8], for info regarding scans to unpassworded SGI accounts. It will illustrate the attempts made to log in to these unpassworded accounts, such as lp, guest, demos, OutOfBox, and EZsetup accounts.

To check the status of all accounts, run:
passwd –a –s

To lock an account you can simply run:
passwd –l <account name>

To force a user to set his/her password at their next logon:
passwd –f <account name>

2.1.3 RUNNING SHADOW

By default shadow passwords are not turned on. It is a good idea to run shadow so only root can access the encrypted passwords. By default, all user passwords are stored in the /etc/passwd file. This file can be read by any user on the system, and as such is a very bad place to store passwords, even encrypted passwords. Password cracking programs such as Crack, can decrypt this file in as little as 15 minutes. To help prevent such easy access, a simple command can be run to convert your /etc/passwd file into /etc/shadow to ensure safer passwords.
pwconv
This program is located in /sbin. Note that you'll have to update /etc/shadow by hand for NIS users.

NIS requires manual entries be added to the /etc/shadow and /etc/passwd files. These entries are usually denoted with a "+" sign in the first character of the username. A "-" sign can also be used if the user once had access, but maybe temporarily left the department and no longer requires access to this system. When they return, you just change the "-" to a "+" and they can once again access the system.

"The shadow file can be served through NIS but that should only be done if the appropriate attributes in **nsd** are set correctly for that map: the **nis_secure** attribute (see **nisserv(7)**) should be turned on and the **mode** attribute (see **nsd(1M)**) should be set to 0700. Failing to

do so introduces a security hole by allowing any user to view entries from the shadow file. This map is not built by default in mdbm_parse. Ypmake needs to be called with the explicit map name **shadow**. Also a line would need to be added to the servers nsswitch.conf file to allow serving the shadow map. An administrator should configure /etc/securenets (see securenets(4)) to list only those hosts that are intended to be NIS clients."[9] IRIX 6.5 – Man Pages, shadow (4)

2.1.4 PASSWORD EXPIRATION

Administrators as well as employees leave companies every day; sometimes switching offices to perform different duties. These are two of the most important reasons for ensuring the passwords on your systems are changed as often as possible. If root's account is hacked, a huge compromise to the system has been made, as well as, a wide open door to possible entry into other systems on the network. This is obviously a bad thing. Password expiration should be set to expire every 30 days for the root account. If someone does have this password or is able to crack the password file, changing the password every 30 days helps prevent a previously authorized user or a "wayward" password from gaining access to your system. Always take into account if someone who has had the root password has left the company, immediate action should be taken to reset the password.

To check to see root's current settings run:
passwd -s root

To set root's account to a 30-day expiration period, run:
passwd -n1 -x30 root

Studies show that if asked, most employees will reveal their password without hesitation. Most others write down their passwords, leaving them under the keyboard, on their desk, or in the drawer. Social behavior is also another method a hacker will use to try and gain passwords. It is vital to system security to ensure you have a policy setting standards for safeguarding passwords. This policy should include as a minimum to keep your passwords kept in a safe place at all times, not to share passwords with others, guidelines for how to create a secure password, minimum length, etc. It should be recommended that all accounts other than root be set to expire every 90 days. This helps to elevate user frustration from frequent password changes, and yet provides at least some protection against a lost password, the user sharing the password, or the hacked password file.

To view status of current settings on all accounts, run:
passwd –a –s

The /etc/default/passwd file is an additional measurement for setting password security. The example below will not allow the user to reset his password within the first week of when he/she initially changed it and it will require them to change it within 13 weeks. Password length must be at least 8 characters.

/etc/default/passwd

```
MAXWEEKS=13
MINWEEKS=1
```

```
PASSLENGTH=8
```

Taken from the IRIX passwd(1) man page:

> "The behavior of the program is influenced by the content of /*etc*/*default*/*passwd* if this file exists. The file is not supplied with the system, but may be locally created and modified as need be. If the file is not present, the default behavior described below is followed. The following item is recognized:
>
> **PASSLENGTH=n**
> minimum length of an acceptable password. This defaults to 6, and has a maximum value of 8."[10] IRIX 6.5 Man Pages, passwd (1.)

2.1.5 Setting strong passwords and validation of the password file

User's and administrator passwords should be strong, good passwords. The most common cracked passwords are those that are found in the dictionary, followed by common names, and user/account names. Try avoiding using these types of passwords. At a minimum your passwords should be at least 8 characters in length and a combination of alpha, numeric and special characters like the "!", "$", "%". Try avoiding the "@" sign since some systems use this as the "ignore" character. This will prohibit you from logging in since your password is ignored after the "@" sign.

"For example, the phrase ''Unix is a trademark of Bell Laboratories'' could give rise to the password ''UiatoBL.'' This essentially creates a password that is a random string of upper and lower case letters. Exhaustively searching this list at 1000 tests per second with only 6 character passwords would take nearly 230 CPU days. Increasing the phrase size to 7 character passwords makes the testing time over 32 CPU *years*"[11] *Klein* "Foiling the Cracker''

"From time to time, you should run the `pwck(1M)` utility to scan the password file. This program reads the file and checks each entry for completeness and notes any inconsistencies. The password checks include validation of:

- the number of fields in each entry
- the login name
- the user ID number
- the group ID number
- the login directory
- the executed program

The default password file to be checked is `/etc/passwd`. If shadow passwords are enabled, the `/etc/shadow` file is checked."[12] "Using pwck to Check the Password File", IRIX Admin: Backup, Security and Accounting, Chapter 4.

2.1.6 SETTING EEPROM PASSWORD

"EEPROM (electrically erasable programmable read-only memory) is user-modifiable read-only memory (ROM) that can be erased and reprogrammed (written to) repeatedly through the application of higher than normal electrical voltage. Only the super-user may alter the **EEPROM** contents." [13] EEPROM Definition

The eeprom password is a very important password to have set on your machine. This password safeguards the internal hardware settings for the system. These settings control from which disk and image to boot your system, the running of diagnostics, and some hardware settings. If left unprotected, these settings may be altered, allowing the system to be booted from a corrupt state and provide "elevated" access to the intruder. Or, they can modify the boot disk or simply remove it, thus causing a denial of service of the system. Both of which require many man-hours to fix and repair.

"If you wish to set your PROM password from within the Command Monitor, perform the following steps:

1. Log in as *root* and shut the system down.
2. When you see this message, press the **Esc** key for the System Maintenance Menu:
   Starting up the system...
   To perform system maintenance instead, press Esc
3. Select option 5 from the System Maintenance Menu to enter the Command Monitor. You see the Command Monitor prompt:
   >>
4. Type the passwd command and press Enter:
   Passwd
5. You see the prompt:
   Enter new password:
6. Enter the password you want for your system and press enter. You see the following prompt:
   Confirm new password:
7. Enter the password again, exactly as you typed it before. If you typed the password the same as the first time, you see the Command Monitor prompt again. Your password is now set. Whenever you access the command Monitor, you will be required to enter this password."

"Note that if you forget your PROM password, but you still know your *root* password, you can reset the PROM password on most systems through the nvram command. If you cannot successfully reset the PROM password, you must remove the PROM or a jumper from your CPU board. See your *Owner's Guide* for information on this procedure.
To assign a new PROM password if you have forgotten it, first clear the existing PROM password from IRIX with the nvram command, and then assign a new one with the passwd command from the PROM monitor.

Clearing the PROM Password Using nvram

To clear the PROM password using the nvram(1M) command, perform the following steps:"[14] Ref: IRIX (r) Admin: Backup, Security, and Accounting, Document Number 007-2862-004

1. Log in as *root*.
2. Give the following command:

```
nvram passwd_key ""
```

2.2 SYSTEM FILES

Comments in system files within the IRIX environment are the same as the UNIX environment. A line beginning with a "#" is a comment. To uncomment the line, simply remove the "#". The directory structure for an Irix system is as follows:

IRIX Files(1): Top Directories[15]

Directory	Purpose	Notes
/	root	contains kernel and top directories
/tmp /usr/tmp	tempory files	Both get used.
/dev	device special files	System device files for identifying devices to system kernel.
/etc	system specific files	Most system config information is contain in /etc and its subdirectories.
/sbin	system administration commands	commands found in /etc in 3.1 moved here and /usr/sbin
/usr/sbin	more system commands	contains a mixture of user and system commands
/var	variable files	e.g. spool, log files, adm files, news, mail, yp databases
/usr/share	shareable files	files that can be shared between systems.
/lib /usr/lib	system libraries	/usr/lib - main location /lib contains small number of vital libs
/usr/etc	system commands and config information	contains largely network related bits of IRIX
/proc	Proc file system. Replaces need to access kernel memory.	Used by ps et al. Not real file system.

The directory structure contains permissions and access to files and directories for three types of users: Owner, Group, and WorldWide or Other. These permissions are in sets of threes: read, write, and execute. On both files and directories they look something like the following:

```
drwxrwxr--        1       root    sysadmin       512     June  5 18:36   /etc
  |     |   |                     |       |
owner |     |                     owner   group
        |   |
    group |
          |
       World
```

It is important for you to understand how these permissions interact when you login to the system. First, you are assigned "permissions" based upon your level of trust. This means that when the administrators setup your account, they gave you a set of permissions via your group ID. For example, if you are a new administrator to the team, you might be added to the "sysadmin" group. You can see by the example above that "sysadmin" is the group of the /etc directory. If you look further to the left, you see the permissions assigned to that group, which are read, write, execute. Those permissions tell the system, that you have full access to the /etc directory and you can execute any "executable" files in that directory. Now let's assume you are just an application user on the system. Your group assignment is

"other". Again using the above example, your group is not "sysadmin" and you certainly aren't logged in as "root", so your login permission set falls into the "World" category. The permissions for "World" are read. So, your login can only "read" inside the /etc directory. That doesn't stop you from executing inside the /etc/ directory, if the permissions assigned to the file to be executed are set with the "x" bit on the "World" permissions. This is relevant to security for the following reason: Let's assume your /etc/passwd file contains the permissions -rw-rw-rw-. This means the owner, group and World have both read and write permissions on the /etc/passwd file. Assuming you logged in as our application user, these permissions would allow you to make changes to the /etc/passwd file. This is a major problem. Only administrators should be able to change these files. You will want to ensure that your permissions on your system critical files are closely scrutinized and tested to ensure functionality of the server and applications. This would be especially true if you have more than one application on a system and each serves a different department. You don't want Department A to access Department B or vice versa. While this is merely an introduction to this topic, it is one often overlooked and misunderstood even for a junior level administrator. I would recommend taking an introductory type course or purchasing a "UNIX for Dummies" type of book for further study of this topic.

2.2.1 Tracking su entries - /etc/default/su and configurations for login - /etc/default/login

All configurations for login will be done via the /etc/default/login file. Below are suggested changes to control the behavior of login.

As an administrator you should always have control over who has access to root's password. Anyone logging in, as root should do so directly through the console and all other users logging in remotely who wish to gain root access will have to log in to their local account and then do a "su". This way you have accountability of who is accessing root's account through 'su'. This is described in detail below in using the /etc/default/su file.

To ensure that root can only log in via the console, make sure the CONSOLE entry in the /etc/default/login file is uncommented.
The entry should look like the following:
CONSOLE=/dev/console
If you'd like to take it a step further, you can strictly enforce that everyone must log in thru their local account and then do a "su", even at the console. Simply said, absolutely no one can log in directly as root. To do this, change the CONSOLE entry to point to /dev/null. The entry will then look like the following:
CONSOLE=/dev/null
There is a downside to this. If all local accounts expire or get locked out, then you will need to boot from CD to recover the login file and change the CONSOLE entry back to /dev/console so that you can log in as root. Or you can recover the password/shadow file to reset the local account password.

All other users logging in remotely who wish to gain root access will have to log in to their local account and then do a "su" to root. Implementing this, especially if the workstation is

in a confined area, is useful for auditing purposes to see who is utilizing the root account remotely.

The /etc/default/su file will keep track of all entries related to a user attempting to perform the 'su' command, whether successful or unsuccessful. To keep a log of these attempts, redirect your SULOG entry to a data file and set your SYSLOG entry to capture ALL attempts. The entry in the /etc/default/su file should look like the following:
SULOG=/var/adm/sulog
SYSLOG=ALL

Successful and unsuccessful attempts made to 'su' will be logged in the sulog and marked with a + or – sign indicating the status of a 'su' attempt. I would recommend not putting any of your logs in / or /etc, since log files are continuously growing, you risk the chance of filling up these partitions and locking up your system. As noted above in the entry, suggest putting log files in /var/adm.

For additional security features, also set the following options in the /etc/default/login file:
PASSREQ =YES --Determines whether all accounts must have passwords. If **YES**, and user has no password, they are prompted for one at login time.
MANDPASS= YES --Like **PASSREQ**, but doesn't allow users with no password to log in.

This ensures all users have passwords set before logging in, if they don't, they will be prompted to set their password immediately upon first login.
As well, you can set the system delay between login failures by modifying the SLEEPTIME entry. The amount of seconds the login is disabled after 3 unsuccessful attempts thru DISABLETIME, as well as MAXTRYS to exit login after 3 unsuccessful attempts. Setting the MAXTRYS parameter to a low number, will introduce a delay (amount of seconds set in the DISABLETIME field, i.e. 30 seconds) in the login process after the failed attempts, thus slowing down the amount of time for a hacker to guess the password. LOGFAILURES will keep track consecutive unsuccessful login attempts from the number you specify, in this case after a user has attempted to unsuccessfully login 3 consecutive times, the attempts will be logged in /var/adm/loginlog. Users with expired passwords are given 2 weeks in which to change their password before they are locked out by setting the IDLEWEEKS entry. The entries should look like the following:

SLEEPTIME=1
DISABLETIME=30
MAXTRYS=3
LOGFAILURES=3
IDLEWEEKS=2

2.2.3 /etc/inetd.conf

Unnecessary services are enabled/started within /etc/inetd.conf. Most of these services have serious vulnerabilities involving buffer overflows or format string issues. Both of which can be exploited to gain root or system level access. To prevent this type of attack, it is best to disable or remove these services. To disable the services, comment the lines containing the

services that are not needed. 'killall –HUP inetd' when you are finished to restart the inet daemon. You may want to disable other unused UDP-based services as well. An alternative, no cost, option to using telnet/ftp/rlogin would be to use Open Secure Shell (OpenSSH). See the following URL for more information:
http://www.openssh.com/[16]
Listed below are services that should be turned off to have a secure environment. You may want to change some of these to meet the configuration for your servers.

As an example for vulnerability information on the buffer overflow in telnet, look at the following URL:
http://www.kb.cert.org/vuls/id/IAFY-4YXSVM[17]

It describes the telnetd program is a server for the telnet remote virtual terminal protocol. There is a remotely exploitable buffer overflow in telnet daemons derived from BSD source code. This vulnerability can crash the server, or be leveraged to gain root access.

```
#telnet
#shell
#login
#exec
#finger
#http
#wn-http
#bootp
#tftp
#ntalk
#tcpmux
#echo
#discard
#chargen
#daytime
#time
#rstatd
#walld
#rusersd
#rquotad
#sprayd
#bootparam
#ypupdated and rexd are somewhat insecure, and not really necessary
#ypupdated
#rexd
#sgi_toolkitbus
#sgi_snoopd
#sgi-esphttp
#ttdbserverd
#tcpmux
#tcpmux/sgi_printer
#tcpmux/sgi_sysadm
```

2.2.4 AUDITING

"The audit subsystem is distributed with your IRIX operating system media, but is not installed by default. To enable auditing, you must use Inst to install the *eoe.sw.audit* software package from your distribution media. Inst is described in detail in *IRIX Admin*: *Software Installation and Licensing*. Once this package has been installed, reboot your system and use the *chkconfig* utility to enable auditing. The chkconfig(1M) reference page provides complete information on the use of *chkconfig* but, simply described, you will see a list of configurable options and a notation of *off* or *on* for each option. The list is in alphabetical order."[18] (sgi techpubs)

If you choose to enable auditing you can do this by running
chkconfig audit on

The *on* option will tell the audit trail to collect records detailing a given event.
The *all* option will collect all event types.
sat_select –on all

To save your current auditing environment while making changes simply do the following:
sat_select –out > /etc/config/sat_select.options

You can restore these configurations at any time by typing:
sat_select `cat /etc/config/sat_select.options`

Your auditing trails are most likely stored in /var/adm/sat. Be sure to keep an eye on the disk space for that filesystem. If you get close to 90% start archiving and cleaning up audit trail files. Refer to Chapter 6 of the Irix Admin Backup, Security and Accounting Document for further information on auditing and different configurations that are available to auditing. Don't forget to change permissions on those files you just created:
chmod 600 /etc/config/sat_select.options
chmod 600 /var/adm/sat

2.2.5 XHOST

Xhost+ is another "convenience" setting that SGI enables. This allows anyone to open windows on your display and even to record what you type at your keyboard.
If the xhost utility is needed, be sure to not use the + (plus) sign. You can specify which systems you would like to give access to by saying "xhost <systemname>". This will allow only that system to utilize the xhost functionality. Using the + sign will allow everyone who has access to your machine to come right in.

2.2.6 WARNING BANNERS

Setup warning banners to let users know what kind of server they are working on, that the server is for authorized users only and that use of the system consents to monitoring. Consult with your corporate attorney for the verbiage to use in these files. This is you first "legal" line of defense in case of intrusion. Usually attorneys like to see a pause in here as well to argue the intruder had to "click" or acknowledge the message before continuing. Since

UNIX based systems don't really allow for that kind of "shell" escape embedded in test, we have to settle for putting in front of them as they login.

Create the /etc/issue file. This banner will be displayed before the login prompt. The contents of the /etc/motd file will be displayed after the user has successfully logged in. Although telnet and ftp are not recommended to be used in this document, there are banner files if you choose to turn those services on. These files are /etc/ftpd and /etc/telnetd. Be sure to chmod 644 on these files to prevent someone from tampering with them.

2.2.7 CRONTAB ENTRIES

The cron is the scheduler on the system. He controls when certain jobs are ran on the system. Any user can be allowed to run scheduled jobs. If you don't have a need for this, it would be very prudent to secure "cron" so that an intruder is not allowed to put a "timebomb" on your system. To accomplish this /etc/cron.d/cron.allow should be created with the users "root" and "sys" only. They both have numerous jobs that require cron to say clean up log files. Other users like "lp" use cron to clean and maintain the print queue. If print is not needed on your system, then removing these entries might help prevent an unauthorized job from running as the "lp" user on your system. Remove all files except "root" and "sys" from /var/spool/cron/crontabs, like 'lp".

Taken from the man pages for Irix:
"Crontab(1) – "If the file /etc/cron.d/cron.allow exists, only users whose names appear in the file are permitted to use crontab. This restriction applies to all users, including root. If that file does not exist, the file /etc/cron.d/cron.deny is checked to determine if the user should be denied access to crontab. If neither file exists, only root is allowed to submit a job. If cron.allow does not exist and cron.deny exists but is empty, global usage is permitted. The allow/deny file consist of one user name per line."[19] IRIX 6.5 Man Pages, crontab(1)

3.0 CLOSING

These are all suggestions for your server. You will need to determine what is applicable to your environment and what is needed to meet the end users needs. For additional information on IRIX support, see http://support.sgi.com/irix or http://techpubs.sgi.com/library/tpl/cgi-bin/browse.cgi?coll=0650&db=man

APPENDIX A

IRIX 6.5.13 Server OS Installation Procedures

These instructions assume an initial install, unless otherwise noted.

Software needed: Foundation 1, 2; Installation Tools and Overlays [3 CDs]; Applications CD

1. Safely power off the system
2. Power on the system, You will notice a window "Running power on diagnostics" followed by another window with a small button labeled "Stop for Maintenance"
3. Click "Stop for Maintenance"
4. Click "Install System Software" icon
5. Click "Local CDROM" icon; click "Install"
6. Insert "Installation Tools and Overlays [1 of 3]" CD; wait for the light to go out on the CD-ROM drive
7. Click "Continue"

If installing over a previous installation, continue. Otherwise, skip to step 8.

A. At the INST> prompt, type "admin"; Press <RETURN>
B. Type "mkfs"; Press <RETURN>
C. Type "y"; Press <RETURN>
D. Type "yes"; Press <RETURN>
E. Type "return"; Press <RETURN> to return to the INST> prompt

8. Eject "Installation Tools and Overlays [1 of 3]" CD
9. Insert "Foundation 1" CD
10. Type "set rulesoverride on"; Press <RETURN> **This will allow for a Core installation**
11. Type "keep *"; Press <RETURN> **This will remove any previously selected software for installation**
12. Type "from /CDROM/dist"; Press <RETURN>
13. Eject the CD; Insert "Foundation 2"
14. Type "open /CDROM/dist"; Press <RETURN>
15. Eject the CD; Insert "Installation Tools and Overlays [1 of 3]
16. Type "open /CDROM/dist"; Press <RETURN>
17. Space through README; When prompted for Maintenance or Feature Stream; Press <RETURN>
18. When prompted, eject the CD; Insert "Overlays [2 of 3]; Press <RETURN>
19. When prompted, eject the CD; Insert "Overlays [3 of 3]; Press <RETURN>
20. When prompted, eject the CD; enter "7" for done; Press <RETURN>
21. Insert "Applications" CD
22. Type "open /CDROM/dist"; Press <RETURN>; Space through README
23. Eject the CD

24. At the INST> prompt, type “go”; Press <RETURN>

When prompted, insert the required CDs:

A. Foundation 1
B. Applications **If an error , type “3” continue; Press <RETURN>**
C. Foundation 2
D. Installation Tools and Overlays [1 of 3]
E. Overlays [3 of 3]

Turn on Auditing:

A. Insert “Foundation 1” CD
B. At the INST> prompt, type “from /CDROM/dist”; Press <RETURN>
C. Type “install eoe.sw.audit”; Press <RETURN>
D. Type “go”; Press <RETURN>

You will be prompted to enter a second CD, Overlays [1 of 3] or Overlays [2 of 2]

25. Type “quit”; Press <RETURN>
26. Type “y” to restart the system; Press <RETURN>

Bibliography

1- Scambray, Joel; McClure, Stuart; Kurtz, George. Hacking Exposed, Network Security Secrets and Solutions, Second Edition.

2 - "Building Blocks for OS Hardening." The Indian Edition, Network Magazine, Solutions for the Competitive Edge. http://www.networkmagazineindiz.com/200110/technology3.htm

3 - http://cert.org (Reference)

4 – http://www.incidents.org (Reference)

5 - http://www.sgi.com/support/security/patches.html

6 - http://support.sgi.com/colls/patches/tools/browse

7 – "System Login and Account Administration / Locking Unused Logins", IRIX Admin: Backup, Security and Accounting, Chapter 4. Irix System Security, Part II. Security. Doc # 007-2862-004. http://techpubs.sgi.com/library/tpl/cgi-bin/getdoc.cgi/srch13@lost%20password/0650/bks/SGI_Admin/books/IA_BakSecAcc/sgi_html/ch04.html (Select System Login and Account Administration and then scroll down to Locking Unused Logins)

8 – CERT Advisory. http://www.cert.org/summaries/CS-98.06.html

9 – IRIX 6.5 – Man Pages, shadow (4). http://techpubs.sgi.com/library/tpl/cgi-bin/getdoc.cgi?coll=0650&db=man&fname=/usr/share/catman/p_man/cat4/shadow.z (Scroll down to NOTES Section)

10 - IRIX 6.5 Man Pages, passwd(1.) http://techpubs.sgi.com/library/tpl/cgi-bin/getdoc.cgi?coll=0620&db=man&fname=/usr/share/catman/u_man/cat1/passwd.z&srch=passwd

11 - *Daniel V. Klein* ''Foiling the Cracker'': A Survey of, and Improvements to, Password Security. Section 2.1. The Survey and Initial Results, paragraph 5. http://packetstorm.decepticons.org/papers/password/ (Select Klein.ps)

12 - "Using pwck to Check the Password File", IRIX Admin: Backup, Security and Accounting, Chapter 4. Irix System Security, Part II. Security. Doc # 007-2862-004. http://techpubs.sgi.com/library/tpl/cgi-bin/getdoc.cgi/srch13@lost%20password/0650/bks/SGI_Admin/books/IA_BakSecAcc/sgi_html/ch04.html#id75080 (Select "Using pwck to Check the Password File)

13 – EEPROM Definition, http://whatis.techtarget.com/definition/0,,sid9_gci213928,00.html

14 - “About PROM Passwords”, IRIX Admin: Backup, Security and Accounting, Chapter 4. Irix System Security, Part II. Security. Doc # 007-2862-004. http://techpubs.sgi.com/library/tpl/cgi-bin/getdoc.cgi/0650/bks/SGI_Admin/books/IA_BakSecAcc/sgi_html/ch04.html#id48777 (Select “Password Administration and then “About PROM Passwords”)

15 – “IRIX Files(1): Top Directories” http://www.dl.ac.uk/TCSC/disco/Courses/IRIXAdmin/sect2/s_2_p4_FileTree.html

16 – OPEN SSH, http://www.openssh.com/ (reference)

17 –CERT Advisory. http://www.kb.cert.org/vuls/id/IAFY-4YXSVM

18 - http://techpubs.sgi.com/library/tpl/cgi-bin/getdoc.cgi/srch8@auditing/0650/bks/SGI_Admin/books/IA_BakSecAcc/sgi_html/ch06.html#id32265

19 – IRIX 6.5 Man Pages, crontab(1) http://techpubs.sgi.com/library/tpl/cgi-bin/getdoc.cgi?coll=0650&db=man&fname=/usr/share/catman/u_man/cat1/crontab.z&srch=crontab