Global Information Assurance Certification Paper

This is the practical assignment for GIAC Certified UNIX Security Administrator (GCUX) Version 1.9 (revised April 8, 2002), practical 6.52.0. I attended the Track 6 seminar at SANS2002 in Orlando, Florida in April of this year. This step-by-step guide conforms to the instructions as per option 1.

This is a step-by-step guide to securing an IRIX Mediabase video web server. It will present: a description of the system, a risk analysis, the preparation of the hard drives, the installation of the operating system, the removal of unused services, the removal of unused applications, the installation of third party applications, the configuring of logging, monitoring, the ongoing daily maintenance, and how to verify the security of the server. It will not only provide a guide to setting up the server, but explanations why certain actions are performed.

Securing computer systems is a business decision. Security should be an integral part of a company's business plan. Security should not hinder a business. Computers are tools that should help employees work smarter and in turn provide a better product or service for their customers. In most cases, computers do not contribute directly to the bottom line, unless of course your business is selling harware or software. The success of the IT Department hinges on the idea that management buys into the idea of securing computers. So it is important that a security director have good people skills. It is a tough job to sell management on spending money for security rather than on research and development of new income producing products or some other project that will generate revenue. A good security officer needs to be more than a geek. They need to see the big picture, be a businessman, and to have a little politician in their personality. The security officer needs to know how to convincingly convey the idea that, “bad things happen to good people.” Companies should be prepared, expect problems, and protect their vital assets, while at the same time focusing on producing profitable products and services.

Securing computer systems is a never-ending battle that continues to change and evolve everyday. The job of a security administrator (technician, help desk rep, instructor, businessman, politician, expediter, and still have a life?) is a challenging task. By sharing our experiences and work practices, we hope to make it a little easier for all of us to protect our computer systems.

Step by Step Securing IRIX Mediabase Server

Dale Drollinger
September 20, 2002

INDEX

I. Description of the System

The hardware includes a Silicon Graphics Origin 200. It has 2 180MHz IP27 processors. The CPU is a MIPS R10000 processor chip revision 3.1. There is 256 Megabytes of main memory, 32 Kilobytes of instruction cache, 32 Kilobytes of data cache, and 2 Megabytes of secondary unified instruction/data cache. 4 integral SCSI controllers, version QL1040B revision 2, work with 17, 18 GB hard drives. The integral Fast Ethernet PCI card runs Full-Duplex at 100 Megabits per second. A CDROM and a 4mm tape drive are installed. The Origin 200 will run IRIX 6.5.16 operating software.

The purpose of this server is to provide on-line computer based training. Training and instructional videos will be stored and cataloged via Mediabase 4.01 database software. There are three general categories of training videos. The categories are computers, management, and engineering. Users can access the server from any internal computer system utilizing their Netscape or Microsoft Internet Explorer browser. Mediabase 4.01 will work in conjunction with the Netscape Fasttrack Web Server version 3.03 to provide high quality interactive real time streaming video. Mediabase requires the XFS file system to guarantee a high rate of I/O in order to support web multicasting and video on demand.

II. Concerns

The video server will be connected to the company intranet. The key security concerns are the never-ending buffer overflows and web based exploits. The video server will have educational information only. There will not be any company information, customer information, employee information, or any mission critical information stored on the server. However, any client that has access to the company intranet automatically has access to video server. It only takes one careless user to download a package that contains a Trojan Horse or bad code to put the entire network at risk. With new exploits coming out daily, access to a machine can be used for other malicious purposes other than just stealing information. The resources from your machine could be used in a distributed denial of service attack. A compromised machine may help an attacker mask where the attack is originating. Trust relationships between systems open holes into the network. An attacker needs just one opening to spring all sorts of havoc on your network as well as outside your network. Security is important in keeping your business running efficiently and effectively.

Proper configuration of the perimeter router and the firewall is the first line of defense to stop intrusions. The perimeter firewall will block spoofed addresses, that is packets from outside the company sourced from internal addresses. They will not allow source-routed packets. No login services from outside our network will be allowed, such as Telnet, ftp, rlogin, NetBIOS, or ssh. RPC, NFS, and ICMP incoming echo requests will be blocked. A firewall will be placed after the perimeter router. Egress filtering rules will also be in place to keep an eye on

outgoing traffic. A Shadow intrusion detection system helps the network administrators keep on top of what is coming and going on their network. Although the intrusion detection system does not prevent anything, it does alert the network engineers to anything out of the norm.

Since it takes only one unknowing user to bypass the firewall, user education is a key to intranet safety. The training would include:

1. How to pick a strong password.
2. Why we should have good passwords.
3. What is the purpose of the company owned computers.
4. What is the accepted use of the company owned computers.
5. Why anything done on the company computer can and will be monitored.
6. Where should users go to download software.
7. Why users should not download software from unauthorized sites.
8. What applications are acceptable.
9. Why applications are unacceptable.
10. What to do if something seems out of the ordinary.
11. What information should or should not be given out, either on the phone or in person.
12. What might be expected when dealing with the help desk or a system administrator.
13. Email practices, including: attachments, spam, hoaxes (Microsoft does not send out email for patches), personal and business usage.
14. Role-playing scenarios to help prepare users for an incident.

The training would be ongoing. There should be an initial training and orientation. Initial training and orientation classes will be held as new employees join the company. After the new employee orientation is completed, each employee needs to sign a security statement. The statement would reiterate the purpose of the companies' computer equipment, that computer use will be monitored, that they have attended the training and orientation class. The signed statement would verify that the employee understands and accepts these conditions. The signed form would be kept in the employee's personnel file in the Human Resources department. At least once a year, all employees should attend security-training sessions. There will be three update classes provided each year. Each class will have a sign in sheet. From the sign in sheet, each employee's personnel file will be updated. It is up to the employee to make sure they attend one session per year. During the employee's annual review, this will be checked and discussed. It is important that employees understand that security is important and that the company computers are tools to help the employees succeed at their jobs. In turn the company will also succeed.

Unneeded services will be shut off, Sendmail, Telnet, and ftp to name a few. The Netscape Fasttrack server has access control lists to limit access to the server. Ipfilter will be used on the video server to provide redundancy and access controls to the server. TCP Wrappers will be used to limit access and improve

our logging capability. There are two user accounts for administrators to log in to the server. Logging on to the server is limited to the console and two other clients. The two clients must use OpenSSH to administer the server.

I will be presenting a method to improve the security on a SGI IRIX server, but this is just the beginning of the work. Log files need to be constantly monitored, tweaks need to be made to configuration files, patches and updates must be installed in a timely manner. User education needs to be monitored, improved and conducted routinely.

The Netscape Fasttrack Server 3.03 has been retired. There are some buffer overflow vulnerabilities that Netscape is not going to patch. They did patch it in their Enterprise Server package. The details of the buffer overflows are in the following: CVE 1999-0744, CVE 1999-0752, CVE 1999-0751, and CVE 1999-0853. This exposure will fall into the category of managed risk. The measures taken to limit our exposure will be presented in more detail in the following pages. The recommendation has been submitted to upgrade the Mediabase software so the open source Apache webserver can be used.

Default operating system installations and initial software builds leave many avenues wide open for intruders to easily compromise your system. Initially, the system has not been patched, software programs may be using default, or even worse, no passwords, well known insecure services like time, chargen, rlogin, and Telnet are running, just to name a few threats. Therefore, the system will not be connected to the network until all the software and configuration hardening is complete.

III. Step by Step Securing IRIX

Here are some conventions that will be followed in the hardening of the video server:

Log in to the console as root to install and configure the video server. After the initial install of the operating system, log in as a regular user and then su to root.
/sbin/su – (Use the full path to su to help prevent someone from placing a bogus su in root's current directory or changing root's path). Using su gives the administrator some accountability as to who did what, when they were logged in as root. This becomes more of an issue when there are numerous administrators. The system will be configured to prevent root logins from anywhere except the console.

IP addresses will not be shown only xxx.xxx.xxx.xxx. Each site would need to use their own addressing scheme.

To secure our UNIX server, some open source applications need to be downloaded from the web. Download software from approved sites. Use MD5 checksums if possible, to assure ourselves that the packages have not been tampered with. Download the applications to an existing system that has already been locked down. Never connect to the web with an open, unpatched, and/or unsecured system! Tar the package up to a 4mm tape. Take the tape over to the new video server and insert it into the 4mm tape drive.

The process would flow as follows:

Log on to your workstation.
Open a browser.
Go to the desired site.
Select the package.
Download it to your home directory.
Close the browser.
cd /home/jsmith
tar cvf /dev/tape openssh-3.4p1.tar.gz
mt –t /dev/tape rewind
mt –t /dev/tape offline
mt –t /dev/tape unload
Insert the tape into the 4mm on the new video server
Log in to the new server.
Go to your home directory.
cd /home/jsmith
tar xvf /dev/tape
mt –t /dev/tape rewind
mt –t /dev/tape offline
mt –t /dev/tape unload
Move the file from your home directory to the installation directory.
We will be using /disk2/admin for many of these freeware utilities.
mv /home/jsmith/openssh-3.4p1.tar.gz /disk2/admin

Follow the usual instructions to unpack the file and read the README and INSTALL files. It is important to read these files to prevent errors, wasting time, and possibly installing it incorrectly, possibly resulting in even more vulnerabilities. The following directions will state where to download the software and then it will start by saying to move the software from your home directory to the installation directory. So the above-mentioned steps will not be repeated each time, but they are necessary.

A. Install Operating System

The current major release of the IRIX operating system is 6.5. SGI releases quarterly updates. The updates ensure stability, reliability, and compatibility. The updates include bug fixes as well as software enhancements. The latest update, IRIX 6.5.16 has four Overlay CD's and one Applications CD.

The CD's are packaged with two manuals, the IRIX 6.5 Installation Instructions and IRIX 6.5.16 Update Guide. It is always a good idea to review these manuals prior to installing the software. This may save you time in the long run by avoiding known "gotchas" or caveats during the install. Don't fall into the trap of, "there is always time to redo something rather, than doing it right the first time!"

1. Partition the Root Disk

We want to partition the disk and then install the operating system.
Turn the power on to the system.
Hit Escape to get the PROM Menu.
The PROM Menu has the following options:

Start System
Install System Software
Run Diagnostics
Recover System
Enter Command Monitor

Insert the IRIX 6.5.16 Installation Tools and Overlays [1 of 4] in the CDROM drive.
Select Enter Command Monitor
Enter the command:
Hinv

This will allow us to record the CD-ROM's controller number, "CDctlr", the CD-ROM's unit number "Cdunit", the system disk's controller number "SYSctlr", and the system disk's unit number "SYSunit". The video server has the CD-ROM controller as 1 and a unit number of 6.

Enter the command:
boot -f dksc(1,6,8)sash64 dksc(1,6,7)stand/fx.64
The fx disk utility program will be launched.
fx: "device-name" = (dksc) enter
fx: ctlr# = 0 enter
fx: drive# = 1 enter
fx: lun# = 0 enter
fx> repartition
fx/repartition> rootdrive
fx/repartition/rootdrive: type of data partition = (xfs) enter
Continue? Yes
fx/repartition> exit

Upon exiting the fx program, you will be returned to the PROM Menu
The drive is ready to put a file system on it.
Select Install System Software
Select Local CD-ROM

A message will display there is no valid file system.
Answer yes to create a file system.
Select a block size of 4096.
Once it completes you will be returned to an Inst> (prompt).

We need to create and mount the /var partition.
Inst>
Enter 11
Admin>
Admin> mkfs /dev/dsk/dks0d1s4 (Creates the file system).
Admin>
Admin> sh (Shell Prompt).
#
mkdir /root/var (Mount point).
mount /dev/dsk/dks0d1s4 /root/var
exit
Enter quit.
Reboot the system.

2. Install Operating System

We need to install the IRIX Operating System Software.
The system will come up and display the PROM Menu.
Select Install System Software.
Select Local CD-ROM.
The INST menu opens up.
Inst>

We need to open up the distribution and overlay CD's. The distribution CD's include:
IRIX 6.5 Foundations 1
IRIX 6.5 Foundations 2
IRIX 6.5 Development Foundations
IRIX 6.5 Development Libraries
The overlay CD's include:
IRIX 6.5.16 Installation Tools and Overlays Disk1
IRIX 6.5.16 Overlays Disk2
IRIX 6.5.16 Overlays Disk3
IRIX 6.5.16 Overlays Disk4
IRIX 6.5.16 Applications Disk

The IRIX 6.5.16 Installation Tools and Overlays Disk 1 is already in the CD-ROM drive. Perform the following commands:
Inst> from /CDROM/dist/
This will load the Overlays Disk 1 into memory. When it completes loading, you will be prompted to load another disk or quit.

Press the eject button on the CD-ROM drive and then insert Overlays Disk 2.
Select /CDROM/dist/
When it completes loading, you will be prompted to load another disk or quit. Press the eject button on the CD-ROM drive and then insert Overlays Disk 3.
Select /CDROM/dist/
When it completes loading, you will be prompted to load another disk or quit. Press the eject button on the CD-ROM drive and then insert Overlays Disk 4.
Select /CDROM/dist/
When it completes loading, you will be prompted to load another disk or quit. Press the eject button on the CD-ROM drive and then insert the Applications Disk.
Select /CDROM/dist/
When it completes loading, you will be prompted to load another disk or quit. Press the eject button on the CD-ROM drive and then insert the Foundations 1 Disk.
Select /CDROM/dist/
When it completes loading, you will be prompted to load another disk or quit. Press the eject button on the CD-ROM drive and then insert the Foundations 2 Disk.
Select /CDROM/dist/
When it completes loading, you will be prompted to load another disk or quit. Press the eject button on the CD-ROM drive and then insert the Development Foundations Disk.
Select /CDROM/dist/dist6.5
When it completes loading, you will be prompted to load another disk or quit. Press the eject button on the CD-ROM drive and then insert the Development Libraries Disk.
Select /CDROM/dist/
Select quit.
This will return you to the Inst >.
Inst>

Enter close at the prompt. This will display all the CD's that were opened. Review the list to make sure we have opened all the required CD's. We will not actually close or exit the menu at this time. Instead, at the prompt we will keep all of the opened CD's.

Inst> keep *
Inst> install standard
Inst> conflicts
This should return, No Conflicts.
Inst> go

The installation starts. Once the software has been installed the Inst prompt will be returned.
Inst quit
It will ask if you want to reboot. Answer yes to reboot.

3. Create Logical Volumes

The system will boot up into multiuser (init 2) mode. We need to prepare the other disks so we can store videos on them. The disks need to be partitioned, a file system needs to be installed, and logical volumes created. Three logical volumes need to be created using 4 disks on each volume. The logical volumes will be called rtmovie2, rtmovie3, and rtmovie4. Two partitions will be created for miscellaneous applications and data storage. They will be called disk2 and opt.

Here are the steps to create logical volume rtmovie2.
Open a terminal window and as root, switch to single user mode.
init 1
Enter the following:
fx -x
The fx disk utility program will be launched.
fx version 6.5, Jul 22, 1998
fx: "device-name" = (dksc)
Select the default (dksc) enter
fx: ctlr# = (0)
Select the default (0) enter
fx: drive# = (1)
Select 6 enter
…opening dksc(0,6,0)
…drive selftest… OK
Select r to repartition the drive
Select o to partition the drive as an optiondrive or a data drive
Select xfs for the partition type
Select /exit

fx -x
The fx disk utility program will be launched.
fx version 6.5, Jul 22, 1998
fx: "device-name" = (dksc)
Select the default (dksc) enter
fx: ctlr# = (0)
Select 2 enter
fx: drive# = (1)
Select 6 enter
…opening dksc(2,6,0)
…drive selftest… OK
Select r to repartition the drive
Select o to partition the drive as an optiondrive or a data drive

Select xfs for the partition type
Select /exit

fx -x
The fx disk utility program will be launched.
fx version 6.5, Jul 22, 1998
fx: “device-name” = (dksc)
Select the default (dksc) enter
fx: ctlr# = (0)
Select 3 enter
fx: drive# = (1)
Select 6 enter
…opening dksc(3,6,0)
…drive selftest… OK
Select r to repartition the drive
Select o to partition the drive as an optiondrive or a data drive
Select xfs for the partition type
Select /exit

fx -x
The fx disk utility program will be launched.
fx version 6.5, Jul 22, 1998
fx: “device-name” = (dksc)
Select the default (dksc) enter
fx: ctlr# = (0)
Select 3 enter
fx: drive# = (1)
Select 3 enter
…opening dksc(3,3,0)
…drive selftest… OK
Select r to repartition the drive
Select o to partition the drive as an optiondrive or a data drive
Select xfs for the partition type
Select /exit

The file system needs to be installed on the partitioned disks.
mkfs /dev/dsk/dks0d6s7
mkfs /dev/dsk/dks2d6s7
mkfs /dev/dsk/dks3d6s7
mkfs /dev/dsk/dks3d3s7

Create a new XLV object using the xlv_make utility.
xlv_make
xlv_make> vol rtmovie2
xlv_make> data
xlv_make> rt

xlv_make> plex
xlv_make> ve dks0d6s7
xlv_make> ve dks2d6s7
xlv_make> ve dks3d6s7
xlv_make> ve dks3d3s7
xlv_make> show
xlv_make> end
xlv_make> exit
Newly created objects will be written to disk.
Is this what you want? (yes)
Enter yes

The above steps need to be repeated for logical volumes, rtmovie3 and rtmovie4.
Logical volume rtmovie3 contains the following:
/dev/rdsk/dks0d5s7
/dev/rdsk/dks2d5s7
/dev/rdsk/dks3d2s7
/dev/rdsk/dks3d5s7

Logical volume rtmovie4 contains the following:
/dev/rdsk/dks0d3s7
/dev/rdsk/dks2d4s7
/dev/rdsk/dks3d1s7
/dev/rdsk/dks3d4s7

4. Create Partition on Data Disk

Create partition disk2.
fx -x
The fx disk utility program will be launched.
fx version 6.5, Jul 22, 1998
fx: “device-name” = (dksc)
Select the default (dksc) enter
fx: ctlr# = (0)
Select default 0 enter
fx: drive# = (1)
Select 2 enter
…opening dksc(0,2,0)
…drive selftest… OK
Select r to repartition the drive
Select o to partition the drive as an optiondrive or a data drive
Select xfs for the partition type
Select /exit

Create partition opt.
fx -x

The fx disk utility program will be launched.
fx version 6.5, Jul 22, 1998
fx: "device-name" = (dksc)
Select the default (dksc) enter
fx: ctlr# = (0)
Select 2 enter
fx: drive# = (1)
Select 3 enter
…opening dksc(2,3,0)
…drive selftest… OK
Select r to repartition the drive
Select o to partition the drive as an optiondrive or a data drive
Select xfs for the partition type
Select /exit

The file system needs to be installed on the partitioned disks.
mkfs /dev/dsk/dks2d3s7
mkfs /dev/dsk/dks0d2s7

Create mount points.
mkdir /disk2
mkdir /opt
mkdir /rtmovie2
mkdir /rtmovie3
mkdir /rtmovie4

5. Mount Filesystems Automatically

We want the file systems to be mounted automatically, so add the mounts to the /etc/fstab file. To protect the binaries in /usr we mount this partition as read only. This could help prevent an intruder from installing a Trojan Horse binary in /usr. Mount /var with the nosuid option set to prevent applications can not change the UID.

```
vi /etc/fstab
/dev/dsk/dks0d1s0       /           xfs     rw,raw=/dev/rdsk/dks0d1s0       0       0
/SWAP750                swap        swap    pri=3   0       0
/dev/dsk/dks0d1s4       /var        xfs     nosuid,rw,raw=/dev/rdsk/dks0d1s4        0  0
/dev/dsk/dks0d1s6       /usr        xfs     ro,raw=/dev/rdsk/dks0d1s4       0  0
/dev/dsk/dks0d2s6       /disk2      xfs     rw,raw=/dev/rdsk/dks0d2s6       0       0
/dev/xlv/rtmovie2       /rtmovie2   xfs     rw      0       0
/dev/xlv/rtmovie3       /rtmovie3   xfs     rw      0       0
/dev/xlv/rtmovie4       /rtmovie4   xfs     rw      0       0
/dev/dsk/dks2d3s2       /opt        xfs     rw      0       0
wq!
```

Shutdown the system.
shutdown –y –g0
The PROM Menu will be displayed.

Start System
Install System Software
Run Diagnostics
Recover System
Enter Command Monitor

Select Start System
The system will boot into multiuser mode.

6. Install Security Patches

Check the SGI website for any security or recommended patches to IRIX 6.5.16. The packages usually come in a tardist format which can be installed from the Inst > prompt.
http://support.sgi.com/support/security/index.html
As of this writing there are some snmp and ftp vulnerabilities. We are running neither of these services, so we will not install any patches.

The steps to install a patch is as follows:
mkdir /tmp/download
mv patch.tardist /tmp/download
cd /tmp/download
tar xvf patch.tardist
inst –f .
Inst> go
On completion, enter quit.
Inst> quit
Remove the files from the temporary download directory.
cd /tmp/download
rm *

B. Configure System

Shutdown the system.
shutdown –y –g0

The PROM Menu will be displayed.

Start System
Install System Software
Run Diagnostics
Recover System
Enter Command Monitor

1. Set IP Address

Select Enter Command Monitor
>

> printenv (Displays the PROM environment variables).
> setenv netaddr xxx.xxx.xxx.xxx (Ensure the video server's ip address is set).

2. Create PROM Password

For added security, there is a PROM password that can be set. It restricts anyone from accessing any of the PROM menu selections except Start System, unless they know the password. From the PROM menu a user could use the fx utility and destroy all the data on the system. Since the system has not booted up, there would not be any logging going on to help see who was on the system. Accessing the cat command in sash, any file on the system could be read. Using miniroot avoids password protection. The user could change the root password without giving it the original password. He could takeover the entire machine and prevent root access even from the administrators. This is the reason a password should be set on the PROM menu. Without knowing the PROM password the user could only start the system up in multiuser mode.

>
> passwd (Enter the password).
> exit (Returns to the PROM menu).
Select Start System.
The system will boot up in multiuser mode.

3. Set resolv.conf

Add domain name and name server to the /etc/resolv.conf file.
vi /etc/resolv.conf
nameserver xxx.xxx.xxx.xxx
domain trainingcompany
Save the file.
chmod 644 /etc/resolv.conf

4. Set sys_id

Save the system name to a file.
videoserver > /etc/sys_id

5. Set nsswitch.conf

Edit the /etc/nsswitch.conf file and set the hosts entry to check files and then dns. Remove all entries with nis. NIS is not used on this system.

vi /etc/nsswitch.conf
hosts: files dns

6. Set hosts file

Edit the /etc/hosts file. The default Internet address is 192.0.2.1 using the hostname IRIS. This will be changed to xxx.xxx.xxx.xxx with the name of

videoserver. Add the two workstations that administration tasks will be performed from. The loopback address must also be in the file.

vi /etc/hosts
xxx.xxx.xxx.xxx learning1 learning1.trainingcompany.com
xxx.xxx.xxx.xxx pc1 pc1.trainingcompany.com
xxx.xxx.xxx.xxx videoserver videoserver.trainingcompany.com
xxx.xxx.xxx.xxx learningcentral \ learningcentral.trainingcompany.com
127.0.0.1 localhost

7. Passwords & User Accounts

Since root can access and do anything on the system, it is extremely important to protect this account with a good password. Select a password that cannot be found in a dictionary, and contains a combination of 8 letters, numbers, and/or characters. Before getting too carried away with a bizzare password, keep in mind you must be able to remember it! Only the administrators for this server as well as the department manager should know the password. The password should be recorded on paper, put in an envelope and stored in a safe. The safe should not be in the same room as the server. The redundancy of people guarantees if the administrator quits, gets injured, or even worse dies, the system is still accessible. Keeping the password in the safe, gives the administrator a back up plan in case he does forget the password. It keeps others from learning the root password. A minimum of people should be able to open the safe. A safe will protect the password in case of a fire. However, if the server burns up, you won't need the password anymore. The password should be changed at 6-month intervals or if one of the users who has been given the password leaves. To change the password for root, enter passwd when logged in as root.

passwd
Enter a password
Re-enter the password

By default passwords are stored in the /etc/passwd file. This is a world readable file. A malicious user could copy the file. The user could take the file to another machine and use a password cracking utility such as "Crack" or "John the Ripper" to reveal the passwords. An administrator would never know that the file was ever copied. Since the cracking would be done elsewhere, the admin would not see any crack processes running. The ps –ef lists all the running processes.

To help prevent this from happening, Shadow Passwords are used. The encrypted passwords are stored in the /etc/shadow file. The pwconv utility is used to initialize shadow passwords. This will start the process of using the /etc/shadow password file. Now the /etc/passwd file only has an X where the password should be. The system knows to search in the /etc/shadow file for password authentication. The shadow file is only readable by root. It will not help

a malicious user if they do make a copy of the /etc/passwd file. They do not have any passwords to crack. The remaining information in the /etc/passwd file however still could provide an intruder with valuable information as to who logs in, what shell, and possibly department names, and phone numbers.

/sbin/pwconv

Create two user accounts for the administrator of the video server.
passmgmt –a –g 20 –s /bin/tcsh jsmith (creates an account in group 20, using T-shell)
/usr/bin/passwd –x 180 –w 10 jsmith (sets password life for 180 days with a 10 day warning to change the password)
mkdir /home/jsmith (create a home directory)
cp /etc/stdlogin /home/jsmith/.login (standard login configuration file for C & T shells)
cp /etc/stdcshrc /home/jsmith/.cshrc (standard csh initialization command for C & T shells)
cp /etc/stdprofile /home/jsmith/.profile (standard login configuration file for Bourne & Korn shells)
chown –R jsmith:user /home/jsmith (change owner from root)
passwd jsmith (give the account a password so the system is not exposed with no password)
New password:
Re-enter new password:
/bin/passwd –f jsmith (forces jsmith to change his password on the next login.)

Do the same for the other administrator account toliver. Do not put "." anywhere in root's or any user's path statement. A Trojan Horse could be placed there and get executed in place of the real command.

Remove unneeded default user accounts. Remove the following from /etc/passwd, /etc/group, and /etc/shadow files:
cmwlogin, diag, uucp, nuucp, Ezsetup, Demos, OutOfBox, 4Dgifts, lp
vi /etc/passwd
go to the line with each of the accounts and enter dd, this deletes the entire line
wq!
Do the same for /etc/group and /etc/shadow.

Set the shell for the following accounts to /dev/null to prevent interactive logins using the following accounts:
daemon
bin
posuser

Use the pwck command to check for incorrect entries in the /etc/passwd file.

/usr/sbin/pwck

If everything checks out OK, nothing should be returned. Correct the error if one does occur.

Passwords are a major form of defense for the video server. One account with a weak or non-existent password not only jeopardizes the video server but every system on the entire network. This is an important topic to be discussed and explained during the user training and orientation sessions.

There are problems with UNIX passwords. The encrypted passwords are stored on the system. Some systems store them in a world readable file others in a file readable only by root. Passwords can be stolen via shoulder surfing. When companies have numerous servers to login to, all requiring different passwords, and some may even force the user to accept automatically generated passwords, this creates another security weakness. One might think that a computer-generated password would fit the bill for a non-dictionary, difficult to crack password. That is true. But it often becomes a problem to the user to remember this password. By the time the user gets familiar with this password it is time to change it again. So what does the user do? They write it down and put it on the bottom of their keyboard, or on the wall, or maybe in the desk drawer. Well so much for good password security. Passwords should be 8 characters in length. The computing power of today's machines doesn't make this length long enough.

The administrator can configure the password length in /etc/default/passwd file.

```
vi /etc/default/passwd
PASSLENGTH=8
wq!
```

Although user passwords are a minor issue on the video server, since there will be just two administrator accounts, they are an important issue in securing UNIX boxes. An alternative method is to use one-time passwords. There is a new password every time. So if a hacker were able to determine your password it would only be good for a short period of time. Some of the drawbacks are implementation issues, user acceptance problems, and operating system upgrades can cause the one time implementation to be overwritten.

There are two freeware products available. They are S/Key and OPIE. They can be downloaded from:

ftp://thumper.bellcore.com/pub/nmh S/Key
http://www.inner.net/opie OPIE

Commercial packages are also available. They are:

SecurID

Defender
SafeWord
CryptoCard
ActivCard

The video server will not use one-time passwords. The administrators should be using strong passwords and following good password etiquette. There will not be any remote logins over unsecured networks outside of the Training Company intranet. Again this falls into the category of managed risk.

8. Set Login Parameters

In the /etc/default/login file we can set the root login option to allow root logins only from the console. Set mandatory passwords for all users to be able to login. Set the maximum tries to prevent someone from guessing at a password indefinitely. We will log to syslog, all login successes and failures (SYSLOG=ALL). We definitely want to know if someone is trying to login as well as anyone who does login. The time to lock out an account if the loginfailures is reached should be set to 30 minutes. The default umask for regular users will be 022. There is a SGI Security Advisory number 20020902-01-I, that references the default umask for root and coredumps. We will set the umask for root to be 077. The root user uses the tcsh shell, so we will modify root's .cshrc file. To check what the root umask is, run the following command:

```
$ su -
 # umask
```

The system should return "022".

Changing the root user's umask to something that will not allow unprivileged
users to read files created by root will result in an extra step to change the permissions (chmod) on files for regular users. Since the only two normal users are administrators, this should not be too much of an inconvenience.

The systune variable is set to 177 for corefiles which prevents regular users from reading core files. We will also be preventing core files from being created when we configure the kernel in the next step (9).

```
vi /etc/default/login
CONSOLE=/dev/console
MANDPASS=YES
MAXTRIES=3
LOGFAILURES=3
SYSLOG=ALL
DISABLETIME=1800
UMASK=022
```

wq!

Add the line umask 077 in root's .cshrc file.
vi /.cshrc
umask 077
wq!

9. Kernel Tuning

Some kernel parameters can be set to help secure our server. Core files can be used to analyze system failures. Core files are generated when an application tries to do something that it is not allowed to do. The operating system sends the application a signal, to kill it. When this occurs the application will create a core file. A core file is an image of the application from memory, and an appropriate identifier, saved to a file named core. Core files are normally world readable and most often take up a large amount of disk space. As of IRIX 6.5.15 a new kernel parameter is set so core files can have a different umask than the default umask of root. This helps to prevent core files from being world readable. The default setting for the systune variable called coremask is 177. Since we set the umask for root to 077, we prevent core files from being read by a regular user. We have covered both bases. The kernel parameter and the umask both prevent world readable core files. They might contain passwords, sensitive information, file contents, and directory structures. An intruder may find lots of valuable information, if he retrieves a core file. A hacker may even attempt to create a core file for the sole purpose of filling up all the available disk space on a file system. This creates a denial of service. Since there is no development going on, on the video server, we are going to limit the size of core files. Developers use core files to understand what happened when an application failed. If the operating system starts crashing, we can always reset this setting until the problem is cleared up and then return it to the restricted setting. Siding with security in mind, we have built in redundancy against core files, zero size, a kernel parameter and an umask preventing users from reading them.

We will set the max size of a core file to 0. It is an extremely large value by default. The initial value was 9223372036854775807.

/usr/sbin/systune rlimit_core_max 0

Eliminate the server from responding to pings sent to the LAN broadcast address. Responding to broadcast pings can be helpful to the network engineers, but it allows a malicious user to use the server as an amplifier in a denial of service attack. The most common broadcast attack is the Smurf attack. The broadcasts are used to map the network. Pings should not be allowed in from outside the firewall.

/usr/sbin/systune ipdirected_broadcast 0

Turn off IP forwarding. This prevents the server from accepting and forwarding packets that are not destined for their local interface address. This turned on; attackers can get around network security measures.

/usr/sbin/systune ipforwarding 0

Turn off ipsendredirects to prevent ICMP redirect packets from redirecting traffic from the server out a different gateway. The ICMP redirect can be used to intercept traffic or to create a denial of service attack.

/usr/sbin/systune ipsendredirects 0

Prevent users from giving away file ownership. Set the restricted_chown paramater to 1. Root should be the only one capable of changing ownership of files.

/usr/sbin/systune restricted_chown 1

The kernel parameters have been changed; now the kernel needs to be reconfigured. After that completes, shutdown and restart the server again.

/etc/autoconfig -f
/etc/shutdown –y –g0

The PROM Menu will be displayed.

Start System
Install System Software
Run Diagnostics
Recover System
Enter Command Monitor

Select Start System

The system will boot up in multiuser mode with the new kernel parameters in effect.

10. Warning Banners

Place a warning banner in the /etc/motd and in /etc/issue. It will also be used in our TCP Wrappers banners.allow file. This will alert anyone who accesses this server that they can be monitored and they should be using the system for company business. A warning banner will help in such cases where termination or prosecution is needed. Besides being included in user training, a user is alerted every time they attempt to log in to the server. A deny banner should also be created to warn anyone trying to logon to the system (TCP Wrappers banners.deny).

Here is the warning banner:

UNAUTHORIZED ACCESS TO THIS COMPUTER SYSTEM AND/OR SOFTWARE IS PROHIBITED.

THE SYSTEM AND DATA CONTAINED THEREIN IS FOR OFFICIAL COMPANY USE ONLY.
The Training Company's video server and related equipment are intended for the communication, transmission, processing and storage of official Training Company information. These systems and equipment are subject to monitoring to ensure proper functioning, to protect against improper or unauthorized use or access, and to verify the presence or performance of applicable security features or procedures, and for other like purposes. Such monitoring may result in the acquisition, recording, and analysis of all data being communicated, transmitted, processed or stored in this system by a user. If monitoring reveals evidence of possible criminal activity, such evidence may be provided to law enforcement personnel. Use of this system constitutes consent to such monitoring. REPORT ALL UNAUTHORIZED USE OR ACCESS TO THE SYSTEM SECURITY OFFICER, (555)666-7777.

Here is the deny banner:

UNAUTHORIZED ACCESS TO THIS TRAINING COMPANY COMPUTER SYSTEM AND/OR SOFTWARE IS PROHIBITED. THIS SYSTEM AND THE DATA CONTAINED THEREIN IS FOR OFFICIAL USE ONLY.
Only registered Training Company machines may access this system. Please contact (555)666-7777 if you feel you have received this message in error.

11. Shut Off Services

Running services that are not required increases the chances of your server being exploited. Unneeded services use up memory and disk space. For example, if the snmp service is not used for network administration, remove it. There have been a few snmp exploits over the past year. Exploits are ususally published on the internet with the step by step details on how to execute them. This allows even the most novice user to launch an attack on your system. So as long as the service is on the system, there is a chance of it getting exploited. So, shut off unused services and if possible remove the product from the server entirely. The following command will prevent the service from starting. Once the desired services are turned off, the server should be rebooted to allow the new settings to take effect.

chkconfig service off

Autofs, automount, pcnfsd, and nfs are NFS services that should be turned off. NFS is a distributed network file sharing protocol developed by Sun Microsystems. It has a history of security problems. NFS is not secure as a default installation. It requires an experienced administrator to configure it correctly. We are not exporting or mounting any directories via NFS. Therefore, all NFS related services should be off.

No printers are attached to the video server so the lp service is off.

Esp is a service that allows remote diagnostics. This is off we do not want anyone to remotely connect to our server. Logins by external vendors or users opens up a whole new avenue of security problems. In most cases you as the administrator, do not have any control over the box that is connecting to your server. Their security policy and procedures may be more lax than what your system is. They may not enforce their policy or they may not even have a policy. This creates a huge hole in your server and network. This should be avoided.

Fontserver, windowsytem, visuallogin, noiconlogin, and xdm are off. X sessions have had numerous exploits and has weak authentication. There are no users, so no one needs to log into an Xsession.

Informix, mediabase, mediad, ns_admin, nss_fasttrack, routed, and xlv are all services that are needed to run Mediabase. These should be set to on.

Ipfilterd allows IP layer packet filtering. Ipfilter demonstrates defense in depth. Although certain traffic should be caught by the firewall, adding additional restrictions at the server may prevent a compromise. The firewall could get compromised. The firewall may get misconfigured after an upgrade. The firewall could fail in an “allow all” state instead of a deny all state. So Ipfilter would be part of our backup plan in case something happens elsewhere on the network. Ipfilter rules can be created in a text based configuration file called ipfilterd.conf.

Savecore allows core files to be saved if the system reboots unexpectedly. This will be set to off. There are no programmers working on this server, which might have a need for core files. We do not want someone who might hack into the system and create a core file to fill up the file system. Core files could contain passwords, file contents, and directory structures. A core file could also use up the disk space causing a denial of service. This is another calculated risk to have core file or not. If the system experiences problems resulting in core files, we would not have them to analyze. If the problem persists, it is easy enough to turn savecore back on. So, we will err in favor of security and set savecore off.

Sshd is on. We will turn this on so the administrators can connect to the video server to do, day-to-day operations. In the ipfilter.conf file we will allow port 22 from the two IP addresses of the administrator’s workstations. OpenSSH uses port 22. Any other attempt to ssh to the video server will be rejected. Fw_sshd will be off preventing IP forwarding to be used.

NIS or NIS+ is not being used so yp, ypmaster, and ypserv are off. NIS has had security issues. It is difficult to manage and configure. Interoperability is a problem. NIS+ is unavailable for IRIX.

Run chkconfig and configure the server as shown below and reboot.

The following is the output of the chkconfig command. This is the output after unneeded services are shutoff.

```
Flag              State
====              =====

acct              off
array             off
autoconfig_ipaddress off
autofs            off
automount         off
cleanpowerdown          on
cpumeter          off
desktop           off
esp               off
fcagent           off
fontserver        off
fw_sshd           off
gated             off
informix          on
ipaliases         off
ipfilterd         on
lockd             off
lp                off
mediabase         on
mediad            on
miser             off
mkpd              on
mmscd             off
mrouted           off
named             off
nds               off
network           on
netwr_client      off
nfs               off
noiconlogin       off
nostickytmp       off
ns_admin          on
nsd               on
nss_fasttrack     on
pcnfsd            off
pmcd              off
privileges        off
proclaim_relayagent  off
proclaim_serveroff
proxymngr         off
quickpage         off
quotacheck        off
quotas            off
rarpd             off
routed            on
rsvpd             off
rtmond            on
rwhod             off
sar               off
```

```
savecore        off
sdpd            off
sendmail        off
sendmail_cf     off
sesdaemon       on
sgi_apache      off
snetd           on
soundscheme     on
ssh             on
sysevent        on
timed           off
timeslave       off
ts              off
verbose         on
videod          off
visuallogin     off
vswap           off
windowsystem    off
xdm             off
xlv             on
yp              off
ypmaster        off
ypserv          off
```

The Internet Daemon, inetd, is the single point of contact for different network services. Since the video server will not have users logging in or accessing files most of the services listed in the /etc/inetd.conf can be removed or commented out.

Services like chargen, discard, daytime, time, and echo are not needed. There are exploits that can cause a denial of service when abusing these services. Telnet, ftp, shell, login, rlogin, and exec are unsecured to run. They allow authentication to go in the clear or to rely on .rhosts or hosts.equiv files to log in. .rhosts files can provide additional machines to attack. Telnet and ftp also provide the operating system version. This could help an attacker focus his attack on exploits known for the newly discovered operating system. Tftpd allows users to transfer files without a password. The administrators to access the server will use OpenSSH. This will encrypt the user name and password as well all data transferred. The OpenSSH configuration file can disable .rhosts support. Finger and rwho can be used to gain user information. The attacker may be able to determine when a user normally is on the system. Rstatd and identd allow users to gain system data. An attacker could find out how a daemon is running, such as root. Ttdbsererd is the tooltalk daemon. It has been plagued with security breaches. It is not used. Remove all the lines from /etc/inetd.conf except the lines starting with sgi_videod and sgi_fam. The file should look like the following:

```
# Internet server configuration database
# $Revision: 1.78 $
sgi_videod/1 stream rpc/tcp wait    root    ?/usr/etc/videod        videod
sgi_fam/1-2 stream rpc/tcp wait    root    ?/usr/etc/fam        fam
```

Comment out unneeded services from the /etc/services file.

vi /etc/services

```
#
# Network services, Internet style
#
# $Revision: 1.44 $
#
tcpmux          1/tcp                   # TCP port multiplexer (RFC 1078)
#echo           7/tcp
#echo           7/udp
#discard        9/tcp           sink null
#discard        9/udp           sink null
#systat         11/tcp          users
#daytime        13/tcp
#daytime        13/udp
#netstat        15/tcp
#qotd           17/tcp          quote
#chargen        19/tcp          ttytst source
#chargen        19/udp          ttytst source
#ftp-data       20/tcp
#ftp            21/tcp
ssh             22/tcp
#telnet         23/tcp
smtp            25/tcp          mail
#time           37/tcp          timserver
#time           37/udp          timserver
rlp             39/udp          resource        # resource location
#name           42/tcp                          # IEN 116
#whois          43/tcp          nicname
#domain         53/tcp          nameserver      # name-domain server
#domain         53/udp          nameserver
#mtp            57/tcp                          # deprecated
#bootp          67/udp          bootps          # bootp server
#bootpc         68/udp                          # bootp client
#tftp           69/udp
#rje            77/tcp          netrjs
#finger         79/tcp
http            80/tcp                          # World-Wide-Web protocol
#link           87/tcp          ttylink
#supdup         95/tcp
#hostnames      101/tcp         hostname        # usually from sri-nic
#iso-tsap       102/tcp
#x400           103/tcp                         # ISO mail
#x400-snd       104/tcp
#csnet-ns       105/tcp
#pop-2          109/tcp                         # Post Office
pop-3           110/tcp
#sunrpc         111/tcp         rpcbind
#sunrpc         111/udp         rpcbind
auth            113/tcp         authentication
#sftp           115/tcp
#uucp-path      117/tcp
#nntp           119/tcp         readnews untp   # USENET News Transfer Protocol
#erpc           121/udp
```

```
ntp             123/udp                         # Network Time Protocol
#loc-srv        135/tcp                         # NCS local location broker
#loc-srv        135/udp
#imap2          143/tcp
#NeWS           144/tcp         news            # Network extensible Window Sys
#snmp           161/udp
#snmp-trap      162/udp                 snmptrap
xdmcp           177/udp                         # X Display Mgr. Control Prot.
#
# UNIX specific services
#
#exec           512/tcp
#biff           512/udp                 comsat
#login          513/tcp
#who            513/udp                 whod
#shell          514/tcp         cmd             # no passwords used
syslog          514/udp
#printer        515/tcp         spooler         # line printer spooler
#talk           517/udp
#ntalk          518/udp
#route          520/udp                 router routed
#timed          525/udp                 timeserver
#tempo          526/tcp         newdate
#courier        530/tcp         rpc
#conference     531/tcp         chat
#netnews        532/tcp         readnews
#netwall        533/udp                         # -for emergency broadcasts
#uucp           540/tcp         uucpd           # uucp daemon
#remotefs       556/tcp         rfs_server rfs  # Brunhoff remote filesystem
#ingreslock     1524/tcp
#
#  ClearCase services
#
#albd           371/udp                         # location broker
# AFS remote authentication
#
ta-rauth 601/tcp            rauth
#
# Kerberos (Project Athena/MIT) services
#
#kerberos       750/udp                 kdc     # Kerberos (server) udp
#kerberos       750/tcp         kdc     # Kerberos (server) tcp
#krbupdate      760/tcp         kreg    # Kerberos registration
#kpasswd        761/tcp         kpwd    # Kerberos "passwd"
#klogin         543/tcp                 # Kerberos rlogin
#nfs            2049/udp        nfs     # Sun NFS
#nfs            2049/tcp        nfs     # Sun NFS
#eklogin        2105/tcp                # Kerberos encrypted rlogin
#kshell         544/tcp         krcmd   # Kerberos remote shell
#x-server       6000/tcp                # X11 window system
#
# The following are not official, so be careful.
#
# SGI demo programs
#sgi-dogfight   5130/udp
#sgi-arena      5131/udp
```

```
#sgi-bznet      5133/udp                # For the BZ demo port
#sgi-objectserver       5135/udp        # SGI Object Server
#sgi-directoryserver    5136/udp        # SGI Directory Server
#sgi-oortnet    5137/udp                # Oort port
#sgi-vroom-server       5138/udp        # SGI Vroom Server
#sgi-vroom-client       5139/udp        # SGI Vroom Client
#sgi-mekton0    5140/udp                # mekton port
#sgi-mekton1    5141/udp                # mekton port
#sgi-mekton2    5142/udp                # mekton port
#sgi-mekton3    5143/udp                # mekton port
#sgi-mekton4    5144/udp                # mekton port
#sgi-mekton5    5145/udp                # mekton port
#sgi-mekton6    5146/udp                # mekton port
#sgi-mekton7    5147/udp                # mekton port
#sgi-pointblank 5150/udp                # pointblank port
#sbm-comm       5555/udp                # Space Boulders port
sgi-dgl         5232/tcp                # SGI Distributed Graphics Lib.
sgi-arrayd      5434/tcp                # SGI array services daemon
realaudio       7070/tcp        ra      # Progr. Tech. RealAudio
wn-http         8778/tcp                # WhatsNew http protocol
#sgi_iphone     32769/tcp               # InPerson phone
sgi_online      1527/tcp                # SGI MediaBase
#xinet-jgui     5969/tcp                # Xinet Java GUI
wq!
```

shutdown –y –g0
At the Prom Monitor select Start System.

12. Remove Files

In addition to stopping services, removing files or entire products from the system, reduces the number of products that could be exploited. It makes the job a little harder for a unwanted user to run a start up script and turn a service back on. A hacker is a determined person. They usually have more time to plan and break into systems than an administrator has administering a system. The intruder can be more focused whereas an administrator always has something else that needs to be done. The administrator should try to make it as difficult as possible for an intruder to get into a system. Quite often a hacker will give up and go on to easier targets, if he runs into too much resistance. Depending on what kind of business you are in, an attacker may not give up easily. It really depends on the motives and objectives of the attacker.

Remove the following packages: snmp, pcnfsd, lp, roboinst, pmie, apache, esp, availmon, and netware. The versions command followed by the product name will show if the product is installed. To remove a product, use the versions command followed by the product name, and remove.

Snmp is not used. It has had numerous vulnerabilities over the years. One of the latest vulnerabilities was reported in SGI Security Advisory, 20020201-010-P, CAN-2002-0013.

versions remove snmp
Pcnfsd allows nfs connections from PC's. NFS is not used. There has been a recent SGI Security Advisory, 20020802-01-I, CERT Advisory CA-1996-08, CAN-1999-0078 on pcnfsd.
versions remove pcnfsd
There are no printers attached to this server.
versions remove lp
Roboinst is an automated package installation tool that is not used.
versions remove roboinst
versions remove pmie
We are not using the Apache webserver.
versions remove apache
Esp is not running, nor do we want any avenue open to allow remote access to the video server.
versions remove esp
Availmon is a set of programs integrated with ESP the system diagnostic software. Since ESP has been removed availmon is not needed.
versions remove availmon
Novell Netware is not run on the video server, so the netware client can be removed.
versions remove netware

The above mentioned products are now uninstalled and removed from the system. Make sure that the start up files are removed from /etc/rc2.d.

/bin/rm /etc/rc2.d/*sgi_apache
/bin/rm /etc/rc2.d/*esp
/bin/rm /etc/rc2.d/*availmon
/bin/rm /etc/rc2.d/*lp
/bin/rm /etc/rc2.d/*roboinst
/bin/rm /etc/rc2.d/*snmp
We do not allow any modems to connect to the video server, so the ppp protocol is not needed.
/bin/rm /etc/rc2.d/*pppstartup
/bin/rm /etc/rc2.d/*netwr_client
/bin/rm /etc/rc2.d/*pcnfsd
/bin/rm /etc/rc2.d/*pmie
mv /usr/etc/fingerd /usr/etc/fingerd.orig (We do not want the finger daemon running)
mv /bin/login /bin/login.orig (We allow ssh connections)
mv /etc/exports /etc/exports.orig (We do not allow any NFS mounts)
touch /etc/exports (Empty exports file)
rm /usr/bsd/rsh (removes the executable for logging in remotely using rsh)
rm /usr/bsd/rcp (removes the executable for copying files using the r commands)

Using xhost and .Xauthority opens up a machine to buffer overflows, denial of service attacks, intruders capturing key strokes, and video output. Although X connections coming from outside our intranet should be blocked at the firewall, we are using the defense in depth strategy. IP Filter should block the IP address. The xdm service is shut off. A X session generally requires a user name and password. There are only two user accounts. X sessions are not needed, so remove the xhost executable.

rm /usr/bin/X11/xhost (Help prevent X server connections via xhost)

13. Configure CRON

Cron allows users to schedule jobs on the system. We do not want anyone except root to be able to do this. Make sure /etc/cron.d/cron.allow and /etc/cron.d/cron.deny do not exist. This configuration allows only the root user to create cron files.

/bin/rm /etc/cron.d/cron.allow
/bin/rm /etc/cron.d/cron.deny

The root crontab uses a umask of 033, which creates log files that are group and other readable. Change all occurrences of 033 to 077.

crontab –e
enter /
enter 033 (this is the text we want to find and replace)
replace 033 with 077
enter n to find the next occurrence
When all the entries have been changed, save the file.
wq!

14. Access Control & IP Filtering

The ipfilter daemon (ipfilterd) filters packets based on their source or destination IP address, the network interface they arrive on, their IP protocol number, their source or destination TCP/UDP port number, or any combination of the above. We will allow SSH traffic on port 22 from two clients: xxx.xxx.xxx.xxx and xxx.xxx.xxx.xxx. All http traffic will be allowed. The RFC 1918 addresses will be blocked: 10.0.0.0, 172.16.0.0, and 192.168.0.0. TCP destination ports 6000 to 6025 are blocked to help prevent X server connections. Other ports are blocked (tcp ports 23, 22, 20, 21, 161, 162, 199, 391, 705, 1993, 6112, udp ports 161, 162 199, 391, 1993, and 6112) based on reports from http://www.incidents.org/ about the top 10 ports being exploited. We have also explicitly rejected ten network addresses of the top ten attackers as per incidents.org. Although these addresses should not be able to reach our server, defense in depth is the best strategy. If an address is not specifically allowed the final rule will reject all tcp traffic. This stance allows just what is needed and rejects the rest. Configure the /etc/ipfilterd.conf file as follows:

```
#
# ipfilterd.conf
# $Revision: 1.3 $
#
# Configuration file for ipfilterd(1M) IP layer packet filtering.
# Lines that begin with # are comments and are ignored.
# Lines begin with a keyword, followed either by a macro definition or
# by an optional interface filter, which may be followed by a protocol filter.
# Both macros and filters use SGI's netsnoop(1M) filter syntax.
#
# The currently supported keywords are:
# accept          : accept all packets matching this filter
# reject : silently discard packets matching this filter
# define: define a new macro to add to the standard netsnoop macros
#
# See the ipfilterd(1M) man page for examples of filters and macros.
#
# The network administrator may find the following macros useful:
#
define ip.netAsrc (src&0xff000000)=$1
define ip.netAdst (dst&0xff000000)=$1
define ip.netBsrc (src&0xffff0000)=$1
define ip.netBdst (dst&0xffff0000)=$1
define ip.netCsrc (src&0xffffff00)=$1
define ip.netCdst (dst&0xffffff00)=$1
define ip.notnetAsrc not((src&0xff000000)=$1)
define ip.notnetAdst not((dst&0xff000000)=$1)
define ip.notnetBsrc not((src&0xffff0000)=$1)
define ip.notnetBdst not((dst&0xffff0000)=$1)
define ip.notnetCsrc not((src&0xffffff00)=$1)
define ip.notnetCdst not((dst&0xffffff00)=$1)
#
# Additional macros:
#
# Filters follow:
#
accept ip.src xxx.xxx.xxx.xxx tcp.dport 22 # administrator's workstation
accept ip.src xxx.xxx.xxx.xxx tcp.dport 22 # administrator's second workstation
reject tcp.dport=23 #rejects all requests going to port 23
reject tcp.dport=22 # ssh cve-2001-0144
reject tcp.dport=20 # ftp data
reject tcp.dport=21 # ftp
reject tcp.dport=161 # snmp cve-2000-0221
reject tcp.dport=162 # snmp trap
reject tcp.dport=199 # smux
reject tcp.dport=391 # snmp
reject tcp.dport=705 # agentx
reject tcp.dport=1993 # Cisco snmp port
reject tcp.dport=6112 # CDE subprocess control cve-2001-0803
reject tcp.dport=6000 # Block X Server Connections
reject tcp.dport=6001 # Block X Server Connections
reject tcp.dport=6002 # Block X Server Connections
reject tcp.dport=6003 # Block X Server Connections
reject tcp.dport=6004 # Block X Server Connections
reject tcp.dport=6005 # Block X Server Connections
```

```
reject udp.dport=161 # snmp cve-2000-0221
reject udp.dport=162 # snmp trap
reject udp.dport=199 # smux
reject udp.dport=391 # snmp
reject udp.dport=1993 # Cisco snmp port
reject udp.dport=6112 # CDE subprocess control cve-2001-0803
accept tcp.dport=http
reject (src&0xFFFF0000)=xxx.xxx.0.0
reject (src&0xFF000000)=10.0.0.0
reject (src&0xFFFF0000)=192.168.0.0
reject (src&0xFFF00000)=172.16.0.0
# The following are the top 10 offenders as of Aug. 8 2002
# http://www.dshield.org/top10.html
reject (src&0xFFF00000)=210.90.32.240
reject (src&0xFFF00000)=210.3.205.65
reject (src&0xFFF00000)=195.170.78.147
reject (src&0xFFF00000)=211.169.240.62
reject (src&0xFFF00000)=80.138.8.192
reject (src&0xFFF00000)=66.250.62.5
reject (src&0xFFF00000)=62.138.192.246
reject (src&0xFFF00000)=64.1.227.131
reject (src&0xFFF00000)=61.102.171.225
reject (src&0xFFF00000)=151.200.221.13
reject tcp
```

Filtering can be configured through Access Control Lists in the Netscape Fasttrack Server. There is an ACL (Access Control List) file that limits the access to the video server. It allows any user from the Training Company's intranet to view media from the video server. It allows the mbase user to administer the Fasttrack Server. It denies anyone from connecting from our intranet from the dial in address (xxx.xxx.xxx.*). It denies all other connections from addresses other than xxx.xxx.*. The following is the ACL file (/var/netscape/fasttrack/httpacl/generated.httpd-videoserver.acl) that controls access to the video server:

```
version 3.0;
acl "path=/var/www/htdocs/mbase/admin";
authenticate (user,group) {
        prompt = "Mediabase Administrator";
};
deny (all)
        user = "anyone";
allow absolute (all)
        user = "mbase";
acl "default";
authenticate (user,group) {
        prompt = "FastTrack Server";
};
deny (all)
 (user = "anyone");

allow (read,execute,info)
 (user = "anyone") and
 (ip = "xxx.xxx.*");
```

```
deny (all)
 (user = "anyone") and
 (ip = "xxx.xxx.xxx.*");

deny (all)
 (user = "anyone") and
 (ip = "193.*");

deny (all)
 (user = "anyone") and
 (ip = "61.*");

deny (all)
 (user = "anyone") and
 (ip = "192.*");

deny (all)
 (user = "anyone") and
 (ip = "195.*");

deny absolute (all)
 (user = "anyone") and
 (ip = "65.*");
```

15. Logging

By default not all logging is turned on. Add the auth.info line to /etc/syslog.conf. Create the /var/adm/loginlog file to capture failed login attempts. Scripts running under cron rotate log files. The daily security script parses some of the log files, compiles the results, saves the results, and mails the results to the root account. Logging information also get sent to the central logging server, learningcentral. By storing log files and the output from security scripts on the local server, the remote server, and via email to another workstation, it becomes very difficult for an intruder to cover his tracks. An intruder may be able to remove some log files on the local host, but he has to dig to find out where everything else is going. These scripts will be discussed later in this paper. The permission of 600 allows read and write access to the root user. Other users do not have a need to know.

```
vi /etc/syslog.conf
auth.info       /var/adm/auth.log
wq!
touch /var/adm/auth.log
chown root:sys /var/adm/auth.log
chmod 600 /var/adm/auth.log
touch /var/adm/loginlog
chown root:sys /var/adm/loginlog
chmod 600 /var/adm/loginlog
```

Here is /etc/syslog.conf

```
# Configuration file for syslogd(1M)
# $Revision: 1.18 $
#
# Formats: selector<TAB>action
#          selector<TAB>filter<TAB>action
kern.debug              |/usr/sbin/klogpp          /var/adm/SYSLOG
*.debug;kern.none       /var/adm/SYSLOG
# Root runs a cron job to rotate the system log file,
# which is very dependent on the format of the following line.
# Therefore, additional loggings at the 'crit' level should only
# be added AFTER the following 'crit' level system logging,
# or the cron job will not work properly.
*.crit                  |/usr/sbin/sysmonpp      /var/adm/SYSLOG
# Enable forwarding to eventmon daemon (UNIX socket)
*.debug                 @@/tmp/.eventmond.events.sock
kern.debug              /usr/adm/kern.log
mail.debug              /usr/spool/mqueue/syslog
mail.debug              /usr/adm/mail.log
daemon.debug            /usr/adm/daemon.log
auth.debug              /usr/adm/auth.log
lpr.debug               /usr/adm/lpr.log
kern.debug              /dev/console
*.debug                 /usr/adm/syslog
*.debug                 /usr/adm/log/syslog
*.info;mail.none /usr/adm/log/syslog
*.debug                 /usr/adm/messages
*.info;mail.none /usr/adm/syslog
*.alert                 /dev/console
*.alert                 root
*.emerg                 *
*.debug                 @learningcentral.trainingcompany.com
```

Here is a copy of the root cron file.

```
# $Revision: 1.48 $
#
# The root crontab can be used to perform accounting data collection
# and cleanup.
#
# Format of lines:
#min    hour    daymo  month   daywk   cmd
#
#
# General SGI practice
#
# Remove old trash
0  5  *   *  *       find / -local -type f '(' -name core -o -name dead.letter ')' -atime +7 -mtime +7 -
exec rm -f '{}' ';'
#
# Remove old vi/ex 'preserved' files
3 5 *    * *       find /var/preserve -local -type f -atime +30 -mtime +30 -exec rm -f '{}' ';'
#
# Rotate the logs
1        1        *        *        0        umask 077;cd /var/cron;if test -s log && test "`/sbin/stat -
qs log`" -ge 10240; then mv -f log OLDlog;touch log; killall 1 cron; fi
```

```
1       1       *       *       0       umask 077;cd /var/adm;if test -s sulog && test
"`/sbin/stat -qs sulog`" -ge 10240; then mv -f sulog OLDsulog;touch sulog; fi
# In order to accept other system loggings at the 'crit' level,
# use only the first 'crit' entry found for log file rotation.
# This works only if system default log file is found first and
# additional 'crit' level logging are added later.
1       1       *       *       0       umask 077;SYSLOGFILE=`grep "\*.crit" /etc/syslog.conf
| awk '$1 != "#" && done == 0 {done =1; print $NF}'`; if test -f $SYSLOGFILE ; then :; else
SYSLOGFILE=/var/adm/SYSLOG; fi;OSYSLOGFILE=`dirname $SYSLOGFILE`\/o`basename
$SYSLOGFILE`;if test -s $SYSLOGFILE && test "`/sbin/stat -qs $SYSLOGFILE`" -ge 10240; then
mv -f $SYSLOGFILE $OSYSLOGFILE;touch $SYSLOGFILE; killall 1 syslogd; fi
#
# If accounting is on it will handle wtmp rotating.
# wtmp and wtmpx are always kept in sync by libc/getut so we should
# always do things to them together
#
2       1       *       *       0       if /etc/chkconfig acct; then :; else umask 077;cd
/var/adm; if test -s wtmp && test "`/sbin/stat -qs wtmp`" -ge 10240; then mv -f wtmp OLDwtmp; mv
-f wtmpx OLDwtmpx; touch wtmp wtmpx; chown adm.adm wtmp wtmpx; fi; fi
# dodisk does the disk accounting
0       2       *       *       4       if /etc/chkconfig acct; then /usr/lib/acct/dodisk >
/var/adm/acct/nite/disklog; fi
#
# Reorganize file systems
#
0 3 *     * 0 if test -x /usr/etc/fsr; then (cd /usr/tmp; /usr/etc/fsr) fi
#
# Repair mangled utmp/wtmp entries
#
1 0       *       *       *       /usr/sbin/chkutent
#
# fsdump updates the local rfind database
#
3 0-3,5-23 * * * /etc/chkconfig rfindd && cd /var/rfindd && exec ./runfsdump
#
# Rotate log file related to amfilter
#
1       1       *       *       0,4     umask 077;SYSLOGFILE=`grep -v "^#" /etc/syslog.conf |
grep amfilter | awk '{print $NF}' 2>/dev/null`; if test "z$SYSLOGFILE" != "z" && test -s
$SYSLOGFILE && test "`/sbin/stat -qs $SYSLOGFILE`" -ge 10240; then
OSYSLOGFILE=`dirname $SYSLOGFILE`\/o`basename $SYSLOGFILE`; mv -f $SYSLOGFILE
$OSYSLOGFILE; touch $SYSLOGFILE; killall 1 syslogd; fi
# MediaBase Services
#
# Rotate /usr/ocs/logs/mbaselog.local files.
1       2       *       *       *       /usr/ocs/sbin/rotatelogs
1       2       *       *       *       /usr/ocs/sbin/rotatelogs -b playlog -p
#
# Back up /usr/sgi/informix/online.log files.
# 0       2       *       *       *       /usr/sgi/informix/bin/backupmsglog
#
# Try to recover shared memory from Informix.
# 0 *     *       *       *       /usr/sgi/informix/bin/shmrecover
# MediaBase Services End
#
# Run John the Ripper quarterly to check for weak passwords
```

```
# runjohn.sh formats the /etc/passwd & /etc/shadow files into a new file
# Testfile
# The /bin/gzcat line takes a pre-computed permuted dictionary and runs
# it against the new Testfile
30      1       1       3,6,9,12        *   /sbin/sh /opt/john-1.6/run/runjohn.sh.sh
#
30 23   *       *       *       /bin/sh /disk2/admin/scripts/daily.script.sh
20 23   *       *       *       /bin/sh /disk2/admin/scripts/chkrootkit.sh
00      *       *       *       *       /usr/lib/sendmail -q
```

16. File & Directory Permissions

File permissions should be changed in some cases to restrict users from accessing or viewing files that they do not have a need to know. Programs like TARA, which we will install and run through cron help administrators with this task. The Solaris Security Step by Step version 2.0 guide also gives guidance to accomplishing this task. Care should be taken when changing directory permissions, applications may cease to work. Be sure to test after making changes.

Change the following:
chmod 600 /etc/fstab
chmod 644 /etc/group
chmod 444 /etc/default/login
chmod 400 /usr/bin/snoop
chmod 400 /var/spoolcron/crontabs/*

Set UID and set GID allows a user to run a file with the permissions of the owner of the file. So this is very risky when this is done with files owned by root. The command: find / -user root –perm –4000 –print will list all set UID files owned by root. Files should be examined closely to see if they are required or not.

Remove the set UID and set GID on the following:
chmod u-s /usr/bin/mail
chmod g-s /usr/bin/mail
chmod u-s /usr/lib/sendmail
chmod g-s /usr/etc/nfsstat
chmod g-s /usr/etc/netstat
chmod g-s /usr/sbin/ipcs
chmod g-s /usr/sbin/movemail
chmod g-s /usr/sbin/mailx

C. Install Software

Installing some applications may requires the ability to write to /usr which is currently mounted as read only. Unmount /usr, install the software, and then reboot the server so that /usr will be mounted as read only again.

```
umount /usr
mount /dev/dsk/dks0d1/s6 /usr
```

1. C Compiler

Install a C compiler. We will be installing and using applications on the video server that require building, compiling and configuring. Install the SGI MIPSpro C Compiler 7.3. Having compilers on a machine adds to the security exposure. If a hacker gets access to the server, he may be able to download some code, and use the on board compilers to compile the application. It might make the task a little more work for the hacker if he has to download and install a compiler to the box. On the other hand, if he has access, and he has downloaded some software already, what is one more package? The more steps the hacker has to take, the more chances there are to slip up. We will be installing the GNC GCC compiler, which includes C and C++. We need the MIPSpro compiler to get the GCC compiler installed. Once this is completed, we could remove the MIPSpro compiler. The GCC compiler will make the job easier for the administrator when compiling other programs such as OpenSSH and SSL. Most freeware programs are written to use the GCC compiler. The caveat here is the tools that make the administrator's life easier can also make the hackers' life easier. Tools can be used for good as well as bad.

```
Inst
Insert Compiler Execution Environment 7.3 CD
Inst> open from /CDROM/dist/dist6.5
Inst> keep *
Inst> install Default
Inst> sh eject
Insert IRIX Development Foundation 1.2
Inst> open from /CDROM/dist/dist6.5
Inst> keep *
Inst> install Default
Inst> sh eject
Insert IRIX 6.5 Development Libraries CD
Inst> open from /CDROM/dist
Inst> keep *
Inst> install Default
Inst> sh eject
Insert the MIPSPro C Compiler 7.3 CD
Inst> open from /CDROM/dist/dist6.5
Inst> keep *
Inst> install Default
Inst> sh eject
```

```
Insert IRIX 6.5.16 Installation Tools and Overlays 1 of 4
Inst> open from /CDROM/dist
Inst> keep *
Inst> install Default
Inst> sh eject
Insert IRIX 6.5.16 Installation Tools and Overlays 2 of 4
Inst> open from /CDROM/dist
Inst> sh eject
Insert IRIX 6.5.16 Installation Tools and Overlays 3 of 4
Inst> open from /CDROM/dist
Inst> sh eject
Insert IRIX 6.5.16 Installation Tools and Overlays 4 of 4
Inst> open from /CDROM/dist
Inst> sh eject
Done
Inst> conflicts
Inst> go
Inst> quit
```

Download the GCC compiler from the SGI freeware site. The download should be performed from another workstation since the video server is not completely configured and locked down.
http://freeware.sgi.com/index-by-alpha.html
This will provide a tardist file that can be installed from the Inst > prompt. The file is fw_gcc-3.0.4.tardist.

```
mkdir /tmp/download
mv fw_gcc-3.0.4.tardist /tmp/download
cd /tmp/download
tar xvf fw_gcc-3.0.4.tardist
inst –f .
Inst> go
On completion, enter quit.
Inst> quit
```

Remove the files from the temporary download directory.

```
cd /tmp/download
rm *
```

2. Sendmail

Download the Sendmail source code from:
ftp://ftp.sendmail.org/pub/sendmail/sendmail.8.12.6.tar.gz

We will not start the Sendmail daemon. A null client configuration will forward mail to a central server for processing. The central mail server is named

letters. This will limit our exposure to vulnerabilities that have plagued the Sendmail application over the years.

```
mv sendmail.8.12.6.tar.gz /disk2/admin
cd /disk2/admin
/usr/sbin/gzcat sendmail.8.12.6.tar.gz | tar xvf –
cd sendmail-8.12.6
Edit the IRIX.6.5 file to use the GCC compiler
cd devtools/OS
vi IRIX.6.5
Change the line define confCC cc to define confCC gcc.
# define confCC cc
define confCC gcc
wq!
cd /disk2/admin/ sendmail-8.12.6
build
cd cf/cf
cp clientproto.mc videoserver.mc
Set the Ostype and Feature parameters in videoserver.mc
vi videoserver.mc
OSTYPE(IRIX6)
FEATURE(nullclient, letters.$m)
/sbin/m4  ../m4/cf.m4  letters.mc > sendmail.cf
cp sendmail.cf  /etc/mail
chown root:sys /etc/mail/sendmail.cf
chmod 644 /etc/mail/sendmail.cf
ln –s /etc/mailsendmail.cf  /etc/sendmail.cf
cd ../../obj.IRIX.6.5.IP22/sendmail
cp sendmail /usr/lib
chown root:sys /usr/lib/sendmail
chmod 4555 /usr/lib/sendmail
Verify that sendmail is off.
The chkconfig command will display the service settings.
chkconfig sendmail off
chkconfig sendmail_cf off
```

Periodically the mail queue should be cleaned of any messages that were not delivered immediately. This should be done through cron. Add the following line to
/var/spool/cron/crontabs/root:

```
crontab –e
00 * * * * /usr/lib/sendmail –q
wq!

ps –ef | grep cron
kill –HUP <cron's ID>
```

```
cd /etc/rc2.d
mv S50mail s50mail
```

3. TCP Wrappers

Download TCP Wrappers from Weiste Venema's FTP site.
ftp://ftp.porcupine.org/pub/security/tcp_wrappers_7.6.tar.gz

Access to services can be filtered by using TCP Wrappers. Two files control the access, hosts.allow and hosts.deny. The administrator specifies what services can be used and by whom. The /etc/hosts.allow is read first to see what is allowed. If it can not find a match the /etc/hosts.deny is consulted. If there is no match in either file access is granted. We should only allow just what is needed by the user community to do their job. So, to accomplish this deny stance, we should allow specific services from specified addresses and deny the rest.

```
mv tcp_wrappers_7.6.tar.gz  /disk2/admin
cd /disk2/admin
gzcat tcp_wrappers_7.6.tar.gz | tar xvf –
cd tcp_wrappers_7.6
chmod 644 Makefile
vi Makefile
Set the Real Daemon Directory, change the FACILITY variable from
LOG_MAIL to LOG_AUTH, and set the CC variable to the GCC compiler.
REAL_DAEMON_DIR=/usr/sbin
FACILITY=LOG_AUTH
CC=/usr/freeware/bin/gcc
make irix6
mkdir /usr/local/sbin
mkdir /usr/local/include
cp safe_finger tcpd tcpdchk try-from /usr/local/sbin
chmod 555 /usr/local/sbin/*
chown root:daemon /usr/local/sbin/*
cp libwrap.a /usr/local/lib
chmod 555 /usr/local/lib/libwrap.a
chown root:daemon /usr/local/lib/libwrap.a
cp tcpd.h /usr/local/include
chmod 444 /usr/local/incude/tcpd.h
chown root:daemon /usr/local/include/tcpd.h
```

Add the services and the hosts to be allowed access. We are only allowing ssh connections from two hosts.

```
vi /etc/hosts.allow
sshd: xxx.xxx.xxx.xxx:  banners /etc/banners.allow
sshd: xxx.xxx.xxx.xxx:  banners /etc/banners.allow
wq!
```

Make the hosts.allow file readable only by root. This would make it harder for unauthorized users to gain information about what network access is granted.

```
chmod 600 /etc/hosts.allow
chown root /etc/hosts.allow
```

Create the deny all statement in the /etc/hosts.deny. We use safe_finger, which comes with TCP Wrappers, instead of the default safefinger. It has been audited for security. Safe_finger attempts to get a long listing of the users who are currently logged in on the source of the connection (@%a) and mails the information to root.

```
vi /etc/hosts.deny
ALL EXCEPT in.fingerd : ALL:  spawn (/usr/bin/safe_finger -l @%a |
/usr/bin/mail -s '%s (%p): connection from %a (%n) '
root@trainingcompany.com) &
wq!
```

Make the hosts.deny file readable only by root. This would make it harder for unauthorized users to gain information about what network access is granted.

```
chmod 600 /etc/hosts.deny
chown root /etc/hosts.deny
```

4. OpenSSH & Prerequisites

OpenSSH will be used to administer the video server from the administrator's workstation. All communications will be encrypted including passwords and data. We will be able to use SCP, a secure way to copy files, and SFTP to securely ftp files to the video server. Telnet and ftp have been disabled due to their inherent security weaknesses, (everything sent in the clear). There are two prerequisites to building OpenSSH. They are OpenSSL and Zlib. Both packages are open source freeware packages. OpenSSL requires Perl to be installed. Perl comes with the default IRIX installation.

Download the OpenSSL source from openssl.org. Revision e corrects the buffer overflow vulnerability reported by CERT, CA-2002-23.
http://www.openssl.org/source/openssl-0.9.6e.tar.gz

```
mv openssl-0.9.6e.tar.gz  /disk2/admin
cd  /disk2/admin
gzcat openssl-0.9.6e.tar.gz | tar xvf –
cd openssl-0.9.6e
./Configure irix-gcc
make
make test
```

make install

Download the Zlib source from freesoftware.com. A double free vulnerability exists in previous versions of zlib. Version 1.1.4 prevents this vulnerability which can cause a denial of service attack. See CERT Advisory CA-2002-07 for further details on this exploit.

ftp://ftp.freesoftware.com/pub/infozip/zlib/zlib-1.1.4.tar.gz

```
mv zlib.tar.gz /disk2/admin
cd /disk2/admin
gzcat zlib.tar.gz | tar xvf –
cd zlib-1.1.4
./configure
vi Makefile
cc=/usr/freeware/bin/gcc
wq!
make
make install
```

Download OpenSSH from www.openssh.com. This version corrects vulnerabilities detailed in CERT Advisory CA-2002-18. The vulnerability allows a user to run arbitrary code as the user running the sshd daemon, most often root. The second vulnerability affects the PAM modules using interactive keyboard authentication. It ignores the challenge response option setting.

```
mv openssh-3.4p1.tar.gz  /disk2/admin
cd /disk2/admin
zcat openssh-3.4p1.tar.gz | tar xvf –
cd openssh-3.4p1
./configure
make
make install
```

Edit the configuration files, sshd_config and ssh_config. We will not allow ssh protocol 1 to connect to the video server since it has security holes. We will not allow .rhosts or RhostsRSAAuthentication. We will only allow two users to ssh, jsmith and toliver. No one can log in as root. The two users should login as regular users and then su to root. This provides trace ability in case of a compromise.

vi /etc/sshd_config

```
# This is ssh server systemwide configuration file.
#
Port 22
Protocol 2
ListenAddress 0.0.0.0
```

```
#ListenAddress ::
HostKey /etc/ssh_host_key
ServerKeyBits 768
LoginGraceTime 600
KeyRegenerationInterval 3600
PermitRootLogin no
#
Don't read ~/.rhosts and ~/.shosts files
IgnoreRhosts yes
Uncomment if you don't trust ~/.ssh/known_hosts for RhostsRSAAuthentication
IgnoreUserKnownHosts yes
StrictModes yes
X11Forwarding no
X11DisplayOffset 10
PrintMotd no
KeepAlive yes

AllowUsers jsmith toliver
# Logging
SyslogFacility DAEMON
LogLevel INFO
#obsoletes QuietMode and FascistLogging

RhostsAuthentication no
#
# For this to work you will also need host keys in /etc/ssh_known_hosts
RhostsRSAAuthentication no
#
RSAAuthentication yes

# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication no
PermitEmptyPasswords no
# Uncomment to disable s/key passwords
#SkeyAuthentication no
#KbdInteractiveAuthentication yes

# To change Kerberos options
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#AFSTokenPassing no
#KerberosTicketCleanup no

# Kerberos TGT Passing does only work with the AFS kaserver
#KerberosTgtPassing yes

CheckMail no
UseLogin no

# Uncomment if you want to enable sftp
Subsystem       sftp    /etc/libexec/sftp-server
MaxStartups 10:30:60
```

wq!

Make the sshd_config file readable only by root. This would make it harder for unauthorized users to gain information about what network access is granted.

chmod 600 /etc/sshd_config
chown root /etc/sshd_config

Edit the /etc/ssh_config file. We want to prevent rsh and RhostsAuthentication due to it's security breeches. Allow protocol 2 only. Protocol 1 has been proven to be insecure.

vi /etc/ssh_config

```
# This is ssh client systemwide configuration file.  This file provides
# defaults for users, and the values can be changed in per-user configuration
# files or on the command line.
# Configuration data is parsed as follows:
#  1. command line options
#  2. user-specific file
#  3. system-wide file
# Any configuration value is only changed the first time it is set.
# Thus, host-specific definitions should be at the beginning of the
# configuration file, and defaults at the end.
# Site-wide defaults for various options
 Host *
 ForwardAgent no
 ForwardX11 no
 RhostsAuthentication no
 RhostsRSAAuthentication no
 RSAAuthentication yes
 PasswordAuthentication yes
 FallBackToRsh no
 UseRsh no
 BatchMode no
 CheckHostIP yes
 StrictHostKeyChecking yes
# IdentityFile ~/.ssh/identity
  IdentityFile ~/.ssh/id_dsa
  Port 22
  Protocol 2
  Cipher blowfish
  EscapeChar ~
```

wq!

Create a start up script so the ssh daemon starts on boot up. This file should be in the /etc/init.d directory. It should be linked to the /etc/rc2.d directory.

cd /etc/init.d
vi ssh

```
#!/bin/sh
case "$1" in
'start')
        if [ -x /usr/local/sbin/sshd –a –f /etc/sshd_config ]; then
```

```
                /usr/local/sbin/sshd –f /etc/sshd_config
        fi
        ;;
'stop')
        kill `cat /etc/sshd.pid`
        ;;
*)
        echo "Usage: $0 { start | stop }"
        ;;
esac
exit 0
        wq!

        chown root:sys /etc/init.d/ssh
        chmod 744 /etc/init.d/ssh
        ln –s /etc/init.d/ssh /etc/rc2.d/S72ssh
```

Each of the administrators should generate a DSA personal key pair. When creating the public/private key pair, the user is prompted for a passphrase. The phrase should be long to help prevent shoulder surfing. This phrase protects your private key. If someone gets the private key they can masquerade as that user. The private key is never transmitted over the network. It woud be a good idea to store the private key on removeable media if possible. Copy the public key to the systems that are going to be connecting with. The public key will be named id_dsa and it will be copied into the authorized_keys2 file on the remote host in the user's .ssh directory under the user's home directory.

Generate a DSA key pair. The user will type the following command and when prompted enter a long passphrase. Make it something you can remember.

```
ssh-keygen –t dsa –b 1024 –f  ~/.ssh/id_dsa
```

Copy the public key to the remote system in the .ssh directory under your home directory.

```
scp  ~/.ssh/id_dsa.pub remotehost:~/.ssh/authorized_keys2
```

5. Fcheck

Download FCheck file integrity program. This is similar to Tripwire. It will check if files have been changed. It can use MD5 checksums. It can be configured to watch specific directories and files. It will be configured to check changes against an original copy stored on a read only CD-ROM disk. The FCheck program is a set of Perl scripts. Once they are unpacked they can be run. No compiling is necessary.

```
mv Fcheck_2.07.59.tar.gz /disk2/admin/scripts
cd /disk2/admin/scripts
gzcat Fcheck_2.07.59.tar.gz | tar xvf –
```

Modify the fcheck program to tell fcheck where the configuration file is.

vi /usr/local/fcheck/fcheck
$config="/usr/local/fcheck/fcheck.cfg";
wq!

mkdir /usr/local/fcheck
mv fcheck /usr/local/fcheck
mv fcheck.cfg /usr/local/fcheck

Here is a copy of the configuration file that fcheck references when running.

/usr/local/fcheck/fcheck.cfg

```
# FCheck.cfg
#
# Directories to be monitored are shown below. Multiple entries may be used
# by using the following 'keyword=variable' format:
#
# [Directory=(path/name)]
# [Directory=(path/name)]
# ...
# If you want recursive direcotry monitoring, place a / at the end of
# the directory name, otherwise the script will interpret the entry as a
# single file or single directory to monitor.
#
# For example the entry "Directory=/usr"
#     will watch everything in the /usr directory
#
# and the entry "Directory=/etc/passwd"
#     will monitor only the password file.
#
# while the entry "Directory=/usr/"
#     will watch everything in the /usr directory, and everything
#     recursively under it, (I.E. /usr/bin..., /usr/local/..., etc.)
Directory          = /tmp
#Directory     = /home/root/scripts/fcheck/
Directory     = /
Directory      = /etc/
Directory     = /bin
Directory     = /opt
Directory     = /opt/MD5/
Directory     = /opt/MIPSpro/
Directory     = /opt/john-1.6/
Directory     = /opt/fcheck/
Directory     = /opt/gzip-1.2.4/
Directory     = /opt/logdaemon-5.6/
Directory     = /opt/modulefiles/
Directory     = /opt/modules/
Directory     = /opt/scripts/
Directory     = /opt/tcp_wrappers_7.6/
Directory     = /lib
Directory     = /sbin/
Directory     = /usr/
Directory     = /var/
```

```
Directory       = /disk2/admin
Directory       = /disk2/admin/tools/
Directory       = /disk2/admin/scripts/

# WARNING
# Use the following exclusions with care,
# only include log files that are constantly updating and are known to
# be written to frequently otherwise you can defeat the purpose of fcheck
# by excluding too much...
#
# Specific files, and/or directories can be excluded.
#
# If used, configure them as full paths and their filenames. Directory
# names must have a "/" appended to the end of its filename in the exclude
# section.
#
Exclusion       = /etc/utmp
Exclusion       = /etc/wtmp
Exclusion       = /rtmovies/
Exclusion       = /rtmovie2/
Exclusion       = /rtmovie3/
Exclusion       = /rtmovie4/
Exclusion       = /disk2/admin/reports/
Exclusion       = /usr/sgi/
Exclusion       = /usr/mbase/
# Miscellaneous settings are passed to fcheck from here.
#
# The "DataBase" keyword points to the absolute path/filename of the baseline
# database file, and is defined next.
#
DataBase        = /usr/local/data/fcheck.dbf
# If you are using a read-only location. You can write the database files to
# one location, and read from an alternate read-only (CD-ROM?) location.
ReadDB          = /cdrom/fcheck.dbf
WriteDB         = /usr/local/data/fcheck.dbf
# Your systems interface for passing messages to its log files, UNIX systems
# are typically found as "/usr/bin/logger".
#
# You could also send messages directly to a line printer if desired.
#
# Win32 platforms are forced to use line printers for now until a error
# logging module is created for NT platforms.
#
#Logger         = /usr/bin/lpr
#
# As of version 2.7.50, you pass logger taglines (-t) options through here.
# Any other options can now be passed to third party loggers, scripts, etc.
Logger          = /usr/bin/logger -tfcheck -hello -w
#AuthLogger     = /usr/bin/logger -tfcheck -pauth.info
#AuthLogger     = /usr/bin/logger -tfcheck -pauth.notice
# This is the system command to determine a files type. Used to determine
# pipes, major/minor numbers.
#
# Only useful on Unix platforms, not portable to Windows (yet?!?!).
FileTyper       = /bin/file
# You may optionally set your hostname from the configuration file if FCheck
```

```
# is unable to determine it on its own.
#
HostName        = "videoserver"
# You may optionally set the system type from the configuration file if
# FCheck is unable to determine it on its own.
# Currently the only accepted option her is "System = DOS", otherwise FCheck
# will default to a UNIX system.
#
#System          = Dos
# This must be set only for readability by you. It in no way effects the scan
# function of FCheck. It only changes what is presented to the end user, so
# the times that are presented to you may not be accurate if not set.
TimeZone        = EST5EDT
# This is used only if you require/desire a hash signature to also be generated
# for each file by use of the '-s' flag. If you do not use the (s)ignature
# flag, then the following variable setting will not impact fcheck in any way.
#$Signature      = /usr/bin/sum
#$Signature      = /usr/bin/cksum
$Signature      = /usr/bin/md5sum
# Include an optional configuration file.
# [CFInclude = (path/config_file_name)]
#CFInclude
# Used for individual file checking (I.E. Possibly FCheck databases!)
#
File       = /usr/local/data/sol.dbf
#
# End of FCheck.cfg file
#
```

6. Cops

Download cops.
http://www.fish.com/cops/cops104+.tar.Z

```
cd /disk2/admin
uncompress cops104+.tar.Z
tar xvf cops104+.tar
cd cops104
./reconfig
make
./cops –v –s . –b bit_bucket
```

7. Network Time Protocol

The NTP (Network Time Protocol) allows the network to keep the system clocks in synch. It is important for numerous reasons such as: scheduling backups, using time based security products, and log file timestamps, possibly for prosecution of intruders.

Download NTP source files from:
http://www.eecis.udel.edu/~ntp/ntp-4.1.1a.tar.gz

mv ntp-4.1.1a.tar.gz /disk2/admin
cd /disk2/admin
gzcat ntp-4.1.1a.tar.gz | tar xvf –
cd ntp-4.1.1a
./configure
make
make install

We will be running as a client machine.
vi /etc/ntp.conf

```
driftfile /etc/ntp.drift
server xxx.xxx.xxx.xxx
server xxx.xxx.xxx.xxx
server xxx.xxx.xxx.xxx
restrict default nomodify
```

wq!

Create a NTP startup script.
Cd /etc/init.d
vi ntp

```
#!/bin/sh
CONFFILE=/etc/ntp.conf

if [ -f $CONFFILE ] ; then
        if [ -x /usr/local/bin/ntpdate ] ; then
                SERVERS=`awk '/^server|peer/ { print $2 } ' \
                $CONFFILE | grep –v ^127`
        /usr/local/bin/ntpdate $SERVERS
        fi
        if [ -x /usr/local/bin/ntpd ] ; then
                echo "Starting NTP."
                /usr/local/bin/ntpd –c $CONFFILE
        fi
fi
```

wq!

chmod 744 /etc/init.d/ntp
chown root:sys /etc/init.d/ntp
ln –s /etc/init.d/ntp /etc/rc2.d/S90ntp
/etc/init.d/ntp

Attach the network cable.
Shutdown the system.
shutdown –y –g0
Start the system back up into multiuser mode.

8. TARA

TARA is a freeware host based vulnerability scanner. It checks for vulnerabilities and configuration errors.

Download from: http://ww-arc.com/tara/index.shtml
mv /disk2/admin/tara-2.0.9
vi Makefile
TIGERHOME=/usr/local/tiger
TIGERBIN=/usr/spool/tiger/bin
TIGERWORK=/usr/spool/tiger/work
TIGERLOGS=/usr/spool/tiger/logs
wq!
make install
cp scripts/check_* /disk2/admin/tara-2.0.9
Edit the scan configuration file tigerrc. The signature database is not up to date so do not run this check. The check embedded script finds too many false positives. Do not run this test either. Do not run crack. We run crack through cron. We only have two user accounts. Turn on the check for world writeable directories.
vi /disk2/admin/tara-2.0.9/tigerrc
Tiger_Check_Signatures=N
Tiger_Check_Embedded=N
Tiger_Run_Crack=N
Tiger_FSScan_WDIR=Y
wq!

9. Chkrootkit

The chkrootkit script is a freeware program that checks for root kits.
Download: http://www.chkrootkit.org/

cp chkrootkit.tar.gz /disk2/admin/scripts
cd /disk2/admin/scripts
gunzip chkrootkit.tar.gz
tar xvf chkrootkit.tar
cd chkrootkit-0.35
make sense

10. John the Ripper Password Checking Tool

Download: http://www.openwall.com/john/

cp john-1.6.tar.gz /opt
cd /opt
gunzip john-1.6.tar.gz
tar xvf john-1.6.tar
cd john-1.6/src
make
make IRIX-gcc

D. Post Installation Tasks

1. Create Backups

Backups are important to ensure the availability of the training materials stored on the videoserver. The entire system should be backed up once a week on Saturday. This will include a full backup as well as a root file system backup. Since the material stored on the videoserver does not change constantly, the once a week schedule should work out fine. Other production systems like development systems need to have their data backed up daily. This would require a different backup plan. A plan would include a full backup once a week, followed by incremental backups each night until the day that full backups are done. A backup would be performed after the quarterly update and any other time the administrator might deem appropriate.

Do not skimp on tapes. You do not want to wear out a tape, only to find out it is not backing up your data when you really need to perform a recovery. To help avoid this, use a different tape for each full backup for the first 10 backups. After the tenth backup reuse the first backup and overwrite the contents. Purchase new tapes at the beginning of each year (January time frame). The same rotation scheme should be used for the root file system backups. The tapes should have a label showing the date of the backup. Tapes should not be left lying around. Once the backup is performed, they should be checked into the media library. This prevents someone from walking by and picking up a copy of your entire system. By storing the tapes elsewhere, you do not keep all your eggs in one basket. If physical damage such as fire or flooding occurs where the server is, it will not affect the backups and vice versa. Checking the tapes into the media library also provides trace ability and accountability.

Log in as a regular user and su to root.
Backup the root file system.
Insert a 4mm tape into the 4mm tape drive.
Enter the following:

```
xfsdump –f /dev/tape –l 0 –v silent /
```

Enter a session label. Example: root514 – This will be a root backup from May 14.

```
root514
```

When it completes, rewind the tape and eject it.

```
mt –t /dev/tape rewind
mt –t /dev/tape offline
mt –t /dev/tape unload
```

Next a full backup is needed. A full backup needs to be run in single user mode.
Enter the following:

```
init 1
```

Insert a 4mm tape into the 4mm tape drive.
Enter:

```
Backup /
```

On completion, rewind the tape and eject it.

```
mt –t /dev/tape rewind
mt –t /dev/tape offline
mt –t /dev/tape unload
```

Bring the system back up to multiuser mode.
Enter:

```
init 2
```

Users should be able to connect to the video server at:
http://videoserver/mbase

2. Create a CD-ROM Disk Toolkit

This CD will contain known good binaries. This can be used if a system has been compromised or thought to have been compromised. Quite often when a system is broken into, a rootkit is installed. The rootkit is a custom piece of software that hides the presence of malicious activity. Common commands like ls, ps, df, netstat, and who are replaced with versions that output results that prevent the administrator from seeing what really is transpiring on the system. The administrator can use the commands from the CD rather than the system ones. Now, the administrator can see who is logged in, see if the network interface is in promiscuous mode, or if there are any connections trying to connect to outside sites. The CD can also be used as a comparison. The administrator can use the diff command to see if the commands on the system are the same as what is on the CD. Beware that the quarterly update can cause the system commands to change, so after an update, the administrator should create a new toolkit CD.

The disk should include:
netstat, lsof, top, ps, ls, diff, su, passwd, netcat, md5checksum, who, w, find, df, rm, mv, cp, chown, chgrp, chmod, tar dd, sh, csh, compress, uncompress, gzip, gunzip, shared libraries, and gcc.

3. Document the Server

After the system has been configured, a binder should be created to document what is installed, how settings are set, and what procedures are used. This can be useful if the system has to be rebuilt or duplicated. If a new administrator is hired, he can get an idea of how the system operates.

Commands can be run to obtain some parameters. The output of the commands can be saved to a file. The files would be collected and printed. Although there is no printer attached to the video server, the results can be moved to another system using a 4mm tape.

Information to be gathered includes:
uname –aR Provides IRIX operating system revision, system name, and processor version.
chkconfig Displays the services that are on and off.
hinv Displays the contents of the system hardware inventory table.
printenv Displays all PROM variables. Once the system is booted up it displays the user's environment.
nvram Displays a list of commands available at PROM.
showprods lists all installed software (installed through the Inst command).
devnm / Displays which disk is the system disk.
prtvtoc –a Displays the partition information for all disks.
/etc/fstab The file lists the automatically mounted filesystems.
df -k Lists the mounted filesystems and their disk usage.
ifconfig – v –a Lists the network interface settings.
ls /etc/rc2.d Lists the start up scripts when booting to multiuser mode.
ls /etc/init.d Lists the start up scripts.
ls –la /dev/dsk Lists the contents of the /dev/dsk directory.
ls –la /dev/rdsk Lists the contents of the /dev/rdsk directory.
/etc/hosts File shows the IP address-hostname database.
/etc/group File shows the groups on the system.
/etc/passwd File shows users, ID, GID, shell, home directory, and information.
/etc/inetd.conf File shows services started by the inetd daemon.
/var/netscape/fasttrack/httpacl/generated.httpd-videoserver.acl ACL for the Netscape Fasttrack Server.
/etc/ipfilterd.conf File contains the ipfiltering rules.
/etc/sshd_config OpenSSH configuration file.
/etc/ssh_config OpenSSH configuration file.
Detail how the sytem can be accessed. Clients can connect to the videoserver and view instructional videos using their browser logged in on any system in the intranet. The administrators can only log in using SSH.
Detail how and when backups should be performed. See section above on backups for details.
Detail how a restore should be performed. See Appendix B.
Detail how the daily security scripts are run and reviewed. See the next section Maintaining Security part 1, Daily Security check of videoserver.
Detail how material to be placed on the video server is approved. See Appendix B.

IV. Maintaining Security

Once a server is configured and hardened, the fun really begins. Activity on the server needs to be monitored. Applications need to be updated or removed depending on the usage. Patches must be kept up to date.

Performance tuning is a never-ending battle. Log files need to be monitored, rotated, and moved offline.

A. Daily Security Check

A daily security script is run through cron every night. The results are saved to files as well as mailed to root. The mail is sent to another system. The script runs several other scripts and then puts the results in one main report. Three other reports are mailed separately from the one total report to the administrator to his workstation. The reports are stored on the server and on the administrator's workstation to provide redundancy. In case of a break in, the intruder might not be able to find all the report files as well as the original log files. The daily reports help the administator understand what is happening on his system and to spot when things are not right. The scripts show different styles of writing scripts. Numerous people have contributed to the maintenance of the scripts. Also as a good UNIX administrator you should follow the mantra of, reuse existing code and modify the code to meet your requirements. Do not reinvent the wheel. Administrators should keep a directory of scripts that they have written or used to accomplish admin tasks, so as new tasks come up, there is a central repository to start from. The following scripts parse log files looking for successful and unsuccessful login attempts, su attempts both successful and unsuccessful, looks for authorization violations, looks at the webserver error logs, looks for .rhosts and hosts.equiv files, looks for changes to files and directories, and looks for root shell scripts. A listing of the scripts is located in Appendix C.

/disk2/admin/scripts/daily.script.sh is the script that gets run by cron. In turn this script calls the following scripts:
/disk2/admin/scripts/su.fails.sh (checks for failed su attempts) You want to know if someone is trying to su to root. Sample output:

```
---------------------------------------------
Begin report for failed su attempts for videoserver on Mon Sep 16 23:30:07 EDT 2002


End report for failed su attempts for videoserver on Mon Sep 16 23:30:07 EDT 2002
---------------------------------------------
```

/disk2/admin/scripts/su.success.sh (checks for successful su attempts) You want to document who is running as root. If something gets changed or misconfigured, you can see which root user did it. It makes it easier to undo the changes, rather than going down some other path looking for the problem. Sample output:

```
---------------------------------------------
Begin report for successful su attempts on videoserver for Mon Sep 16 23:30:07 EDT 2002

SU 09/16 12:54 + ttyq0 toliver-root

End report for successful su attempts on videoserver for Mon Sep 16 23:30:07 EDT 2002
```

/disk2/admin/scripts/logins.sh (checks for successful login attempts) You want to know who logs in to the videoserver. It should only have two user accounts. Sample output:

Begin report of valid logins for videoserver on Mon Sep 16 23:30:07 EDT 2002

toliver ttyq0   learning1   Mon Sep 16 12:54 - 15:07 (02:12)

End report of valid logins for videoserver on Mon Sep 16 23:30:07 EDT 2002

/disk2/admin/scripts/login.fails.sh (checks for failed login attempts) You want to know if someone is trying to guess usernames and passwords.

Begin report of failed login attempts (5 conseq) for videoserver on Mon Sep 16 23:30:07 EDT 2002

End report of failed login attempts for videoserver on Mon Sep 16 23:30:07 EDT 2002

/disk2/admin/scripts/find_rhosts.sh (checks for .rhosts files) .rhosts files should be removed from the system. They are a security threat, in that they allow a user to log on without a password. OpenSSH is avilable, so there is no reason to use .rhosts files. This removes any.rhosts files. Sample output:

Begin report of find_rhosts for videoserver on Mon Sep 16 23:30:07 EDT 2002

End report of find_rhosts for videoserver on Mon Sep 16 23:30:07 EDT 2002

/disk2/admin/scripts/find_hostsequiv.sh (checks for hosts.equiv files) Hosts.equiv files are basically the same as .rhosts files. This removes any hosts.equiv files.

Begin report of /etc/hosts.equiv for videoserver on Mon Sep 16 23:31:01 EDT 2002

End report of /etc/hosts.equiv for videoserver on Mon Sep 16 23:31:01 EDT 2002

/disk2/admin/scripts/find_root_sh_scripts.sh (checks for root shell scripts) You do not want to find any root kits on your system. Root kits normally have programs to hide processes, clean log files, ftp or even worse ssh out to malicious sites. If you find a root kit it is time to follow the Computer Security Incident Handling Step-by-Step book from SANS.

Begin report of find_root_sh_scripts for videoserver on Mon Sep 16 23:31:01 EDT 2002

End report of find_root_sh_scripts for videoserver on Mon Sep 16 23:31:01 EDT 2002

/disk2/admin/scripts/auth.log.sh (checks /var/adm/auth.log file) Sample output:

Begin report for auth.log on videoserver for Mon Sep 16 23:31:58 EDT 2002

Sep 16 12:54:24 6E:videoserver sshd[1595977]: Accepted password for toliver from xxx.xxx.xxx.xxx port 1139 ssh2
Sep 16 12:54:24 6E:videoserver sshd(pam_unix)[1599042]: session opened for user toliver by (uid=11111)
Sep 16 12:54:36 5E:videoserver su[1598609]: succeeded: ttyq0 changing from toliver to root
Sep 16 12:59:19 6E:videoserver sshd[1599072]: Accepted password for toliver from xxx.xxx.xxx.xxx port 1146
Sep 16 12:59:19 6E:videoserver sshd(pam_unix)[1599063]: session opened for user toliver by (uid=11111)
Sep 16 13:00:30 6E:videoserver sshd(pam_unix)[1599063]: session closed for user toliver
Sep 16 15:07:21 6E:videoserver sshd[1599042]: Received disconnect from xxx.xxx.xxx.xxx: 11: All open channels closed
Sep 16 15:07:21 6E:videoserver sshd(pam_unix)[1599042]: session closed for user toliver

End report for auth.log on videoserver for Mon Sep 16 23:31:58 EDT 2002

/disk2/admin/scripts/webserver.error.log.sh (checks the error log file for the webserver) This looks for intruders trying to exploit your webserver. Sample output from a saint scan of the videoserver:

Begin report of Webserver Errors for videoserver on Thu Sep 5 23:31:59 EDT 2002

[05/Sep/2002:11:08:46] warning: for host xxx.xxx.xxx.xxx trying to GET /cgi-bin/n0nexi5tent_fi1e.pl, send-cgi reports: cannot find CGI program /var/www/cgi-bin/n0nexi5tent_fi1e.pl (No such file or directory)

[05/Sep/2002:11:08:46] warning: for host xxx.xxx.xxx.xxx trying to GET /cgi-bin/n0nexi5tent_cgi, send-cgi reports: cannot find CGI program /var/www/cgi-bin/n0nexi5tent_cgi (No such file or directory)

[05/Sep/2002:11:08:46] warning: for host xxx.xxx.xxx.xxx trying to GET /n0nexi5tent_fi1e.html, send-file reports: can't find /var/www/htdocs/n0nexi5tent_fi1e.html (No such file or directory)

[05/Sep/2002:11:08:46] warning: for host xxx.xxx.xxx.xxx trying to GET /\../\../\../\../\../etc/group, send-file reports: can't find /var/www/htdocs/\../\../\../\../\../etc/group (No such file or directory)

[05/Sep/2002:11:08:46] warning: for host xxx.xxx.xxx.xxx trying to GET /cgi-bin/webdist.cgi, send-cgi reports: cannot find CGI program /var/www/cgi-bin/webdist.cgi (No such file or directory)

[05/Sep/2002:11:08:46] warning: for host xxx.xxx.xxx.xxx trying to GET /directory.php, send-file reports: can't find /var/www/htdocs/directory.php (No such file or directory)

[05/Sep/2002:11:08:46] warning: for host xxx.xxx.xxx.xxx trying to GET /cgi-bin/campas, send-cgi reports: cannot find CGI program /var/www/cgi-bin/campas (No such file or directory)

[05/Sep/2002:11:08:47] warning: for host xxx.xxx.xxx.xxx trying to GET /cgi-bin/sewse, send-cgi reports: cannot find CGI program /var/www/cgi-bin/sewse (No such file or directory)

[05/Sep/2002:11:08:47] warning: for host xxx.xxx.xxx.xxx trying to GET /cgi-script/CSNews/CSNews.cgi, send-cgi reports: cannot find CGI program /var/www/htdocs/cgi-script/CSNews/CSNews.cgi (No such file or directory)

End report of Webserver Errrors for videoserver on Mon Sep 5 23:30:07 EDT 2002

/disk2/admin/scripts/md5binchk.sh (checks for changes in /usr/bin) It does the same as the fcheck file integrity program. It is an added measure of redundancy. Sample output:

Begin report of MD5 Checksum Check for videoserver on Mon Sep 16 23:31:59 EDT 2002

End report of MD5 Checksums for videoserver on Mon Sep 16 23:31:59 EDT 2002

/disk2/admin/scripts/dfout.sh (checks file system usage) You do not want a file system to fill up and cause a denial of service.

Begin report of File Structures on videoserver on Mon Sep 16 23:31:59 EDT 2002

```
Filesystem          Type  kbytes    use     avail  %use Mounted on
/dev/root           xfs  3913932  3426004   487928  88  /
/dev/dsk/dks0d1s4               xfs  1111000   688820   422180  62  /var
/dev/dsk/dks2d3s2     xfs  2043328    53048  1990280   3  /opt
/dev/dsk/dks0d2s6     xfs 13548400   661628 12886772   5  /disk2
/dev/xlv/rtmovie2     xfs 71111680  7176692 63934988  11  /rtmovie2
/dev/xlv/rtmovie4     xfs 71111680     2176 71109504   1  /rtmovie4
/dev/xlv/rtmovie3     xfs 71111680     2176 71109504   1  /rtmovie3
```

End report of File Structures on videoserver on Mon Sep 16 23:31:59 EDT 2002

/disk2/admin/scripts/finduid.pl (checks for any UID's of 0 besides the root account) You want to know if someone has added a user account with UID 0, which would make the user in effect root! Sample output:

Begin report for UID 0 on videoserver on Mon Sep 16 23:31:59 EDT 2002

End report for UID 0 on videoserver on Mon Sep 16 23:31:59 EDT 2002

/usr/local/fcheck/fcheck (checks for changes to specified files & directories) It uses MD5 checksums. It also stores a copy of the original checksums on a CD-ROM disk. You can always run a diff from the original CD and what is currently installed. The daily script does store a copy of the original on the server. Sample output:

/usr/local/fcheck/fcheck.cfg
- -

PROGRESS: validating integrity of Files
STATUS: No changes on "videoserver" to: /usr/local/data/sol.dbf
passed...
PROGRESS: validating integrity of /tmp
STATUS:passed...
PROGRESS: validating integrity of /
STATUS:No changes on "videoserver" to: /
passed...
PROGRESS: validating integrity of /etc/banners.deny/
STATUS:passed...
PROGRESS: validating integrity of /etc/config/
STATUS:passed...
PROGRESS: validating integrity of /etc/default/
STATUS:passed...
PROGRESS: validating integrity of /etc/init.d/
STATUS:passed...
PROGRESS: validating integrity of /etc/net/
STATUS:passed...
PROGRESS: validating integrity of /etc/rc0.d/
STATUS:passed...
PROGRESS: validating integrity of /etc/rc2.d/
STATUS:passed...
PROGRESS: validating integrity of /etc/tt/
STATUS:passed...
PROGRESS: validating integrity of /bin
STATUS:passed...
PROGRESS: validating integrity of /opt
STATUS:passed...
PROGRESS: validating integrity of /opt/MD5/
STATUS:passed...
PROGRESS: validating integrity of /opt/MIPSpro/

```
STATUS:passed...
PROGRESS: validating integrity of /opt/john-1.6/
STATUS:passed...
PROGRESS: validating integrity of /opt/fcheck/
STATUS:passed...
PROGRESS: validating integrity of /opt/gzip-1.2.4/
STATUS:passed...
PROGRESS: validating integrity of /opt/logdaemon-5.6/
STATUS:passed...
PROGRESS: validating integrity of /opt/modulefiles/
STATUS:passed...
PROGRESS: validating integrity of /opt/modules/
STATUS:passed...
PROGRESS: validating integrity of /lib/
STATUS:passed...
PROGRESS: validating integrity of /sbin/
STATUS:passed...
PROGRESS: validating integrity of /usr
STATUS:passed...
PROGRESS: validating integrity of /usr/Cadmin/
STATUS:passed...
PROGRESS: validating integrity of /usr/CaseVision/
STATUS:passed...
PROGRESS: validating integrity of /usr/CosmoPlayer/
STATUS:passed...
PROGRESS: validating integrity of /usr/Imgtcl/
STATUS:passed...
PROGRESS: validating integrity of /usr/Motif-1.2/
STATUS:passed...
PROGRESS: validating integrity of /usr/Motif-2.1/
STATUS:passed...
PROGRESS: validating integrity of /usr/NetVis/
STATUS:passed...
PROGRESS: validating integrity of /usr/SVP/
STATUS:passed...
PROGRESS: validating integrity of /usr/SpeedShop/
STATUS:passed...
PROGRESS: validating integrity of /usr/ToolTalk/
STATUS:passed...
PROGRESS: validating integrity of /usr/WebFace/
STATUS:passed...
PROGRESS: validating integrity of /usr/WorkShop/
STATUS:passed...
PROGRESS: validating integrity of /usr/WorkShopMPF/
STATUS:passed...
PROGRESS: validating integrity of /usr/adobe/
STATUS:passed...
PROGRESS: validating integrity of /usr/bin/
STATUS:passed...
PROGRESS: validating integrity of /usr/bsd/
STATUS:passed...
PROGRESS: validating integrity of /usr/cpu/
STATUS:passed...
PROGRESS: validating integrity of /usr/custlink/
STATUS:passed...
PROGRESS: validating integrity of /usr/diags/
```

```
STATUS:passed...
PROGRESS: validating integrity of /usr/disktest/
STATUS:passed...
PROGRESS: validating integrity of /usr/etc/
STATUS:passed...
PROGRESS: validating integrity of /usr/freeware/
STATUS:passed...
PROGRESS: validating integrity of /usr/gfx/
STATUS:passed...
PROGRESS: validating integrity of /usr/include/
STATUS:passed...
PROGRESS: validating integrity of /usr/info/
STATUS:passed...
PROGRESS: validating integrity of /usr/java/
STATUS:passed...
PROGRESS: validating integrity of /usr/lib/
STATUS:passed...
PROGRESS: validating integrity of /usr/lib32/
STATUS:passed...
PROGRESS: validating integrity of /usr/lib64/
STATUS:passed...
PROGRESS: validating integrity of /usr/local/
STATUS:
        WARNING: ["videoserver"] /usr/local/data/fcheck.dbf
        [Sizes: 2113536 - 5111717, Times: Sep 19 00:26 2002 - Sep 19 00:41 2002]
PROGRESS: validating integrity of /usr/nds/
STATUS:passed...
PROGRESS: validating integrity of /usr/ns-home/
STATUS:passed...
PROGRESS: validating integrity of /usr/pcp/
STATUS:passed...
PROGRESS: validating integrity of /usr/people/
STATUS:passed...
PROGRESS: validating integrity of /usr/relnotes/
STATUS:passed...
PROGRESS: validating integrity of /usr/sbin/
STATUS:passed...
PROGRESS: validating integrity of /usr/sgitcl/
STATUS:passed...
PROGRESS: validating integrity of /usr/share/
STATUS:passed...
PROGRESS: validating integrity of /usr/sitemgr/
STATUS:passed...
PROGRESS: validating integrity of /usr/sysadm/
STATUS:passed...
PROGRESS: validating integrity of /usr/var/
STATUS:passed...
PROGRESS: validating integrity of /usr/webdocs/
STATUS:passed...
PROGRESS: validating integrity of /var
STATUS:passed...
PROGRESS: validating integrity of /disk2/admin
STATUS:passed...
PROGRESS: validating integrity of /disk2/admin/tools/
STATUS:passed...
PROGRESS: validating integrity of /disk2/admin/scripts/
```

STATUS:passed...

/disk2/admin/cops_104/cops (runs daily cops report) This is an old program but still can provide some redundancy. Sample output:

```
ATTENTION:
Security Report for Thu Sep 19 00:12:03 EDT 2002
from host videoserver

**** root.chk ****
**** dev.chk ****
**** is_able.chk ****
**** rc.chk ****
**** cron.chk ****
**** group.chk ****
**** home.chk ****
**** passwd.chk ****
Warning!  Password file, line 16, invalid login directory:
        posuser:x:55555:20:::/dev/null
**** user.chk ****
**** misc.chk ****
**** ftp.chk ****
**** pass.chk ****
**** kuang ****
**** bug.chk ****
```

/disk2/admin/tara/tiger Looks for vulnerabilities and configuration errors, it is good for finding world writeable directories. Sample output:

```
Security scripts *** 2.0.9 ARC, 1999.0907.2100 ***
Tue May 28 11:24:05 EDT 2002
11:24> Beginning security report for videoserver (IP27 IRIX 6.5).

# Performing check of passwd files...

# Performing check of group files...
# Performing check of user accounts...
# Checking accounts from /etc/passwd.

# Performing check of /etc/hosts.equiv and .rhosts files...

# Checking accounts from /etc/passwd...

# Performing check of .netrc files...

# Checking accounts from /etc/passwd...

# Performing check of /etc/default/login, /securetty, and /etc/ttytab...


# Performing check of PATH components...
# Only checking user 'root'

# Performing check of anonymous FTP...

# Performing checks of mail aliases...
```

```
# Checking aliases from /usr/lib/aliases.

# Performing check of `cron' entries...
--WARN-- CRON file `' is owned by adm.
--WARN-- CRON file `' is owned by sys.
--WARN-- CRON file `' is owned by sys.
--WARN-- CRON file `' is owned by sys.
--WARN-- CRON file `' is owned by sys.

# Performing check of 'services' and 'inetd'...
# Checking services from /etc/services.
# Checking inetd entries from /usr/etc/inetd.conf

# Performing NFS exports check...

# Performing check of system file permissions...
--WARN-- [perm001w] The owner of /var/tmp should be root (owned by sys).
--WARN-- [perm001w] /dev/vme should not have owner search.
--WARN-- [perm001w] /dev/vme should not have group read.
--WARN-- [perm001w] /dev/vme should not have group write.
--WARN-- [perm001w] /dev/vme should not have group search.
--WARN-- [perm001w] /dev/vme should not have world read.
--WARN-- [perm001w] /dev/vme should not have world search.
# Checking for known intrusion signs...

# Performing check of files in system mail spool...

# Performing system specific checks...
# Running './scripts/check_sendmail'...

# Checking sendmail...

# Checking unusual file names...

# Looking for unusual device files...

# Checking symbolic links...
```

/dev/vme must keep the permission of 755 for Mediabase to function correctly.

/disk2/admin/scripts/chkrootkit.sh The chkrootkit script is a freeware program that checks for root kits. It is not executed from the daily security script, but the output of the script is Emailed to root via the daily security script. chkrootkit.sh is started daily from cron. Sample output:

```
ROOTDIR is `/'
Checking `amd'... not found
Checking `basename'... not infected
Checking `biff'... not found
Checking `chfn'... not infected
Checking `chsh'... not infected
Checking `cron'... not infected
Checking `date'... not infected
Checking `du'... not infected
Checking `dirname'... not infected
Checking `echo'... not infected
Checking `egrep'... not infected
```

```
Checking `env'... not infected
Checking `find'... not infected
Checking `fingerd'... not infected
Checking `gpm'... not found
Checking `grep'... not infected
Checking `hdparm'... not found
Checking `su'... not infected
Checking `ifconfig'... not infected
Checking `inetd'... not infected
Checking `inetdconf'... not infected
Checking `identd'... not found
Checking `killall'... not infected
Checking `ldsopreload'... not tested
Checking `ls'... not infected
Checking `lsof'... not found
Checking `mingetty'... not found
Checking `netstat'... not infected
Checking `named'... not found
Checking `passwd'... not infected
Checking `pidof'... not found
Checking `pop2'... not found
Checking `pop3'... not found
Checking `ps'... not infected
Checking `pstree'... not found
Checking `rpcinfo'... not infected
Checking `rlogind'... not infected
Checking `rshd'... not infected
Checking `slogin'... not found
Checking `sendmail'... not infected
Checking `sshd'... not infected
Checking `syslogd'... not infected
Checking `tar'... not infected
Checking `tcpd'... not infected
Checking `top'... not infected
Checking `telnetd'... not infected
Checking `timed'... not infected
Checking `traceroute'... not infected
Checking `write'... not infected
Checking `aliens'... no suspect files
Searching for sniffer's logs, it may take a while... nothing found
Searching for HiDrootkit's default dir... nothing found
Searching for t0rn's default files and dirs... nothing found
Searching for t0rn's v8 defaults... nothing found
Searching for Lion Worm default files and dirs... nothing found
Searching for RSHA's default files and dir... nothing found
Searching for RH-Sharpe's default files... nothing found
Searching for Ambient's rootkit (ark) default files and dirs... nothing found
Searching for suspicious files and dirs, it may take a while...
Searching for LPD Worm files and dirs... nothing found
Searching for Ramen Worm files and dirs... nothing found
Searching for Maniac files and dirs... nothing found
Searching for RK17 files and dirs... nothing found
Searching for Ducoci rootkit... nothing found
Searching for Adore Worm... nothing found
Searching for ShitC Worm... nothing found
Searching for Omega Worm... nothing found
```

```
Searching for Sadmind/IIS Worm... nothing found
Searching for MonKit... nothing found
Searching for anomalies in shell history files... nothing found
Checking `asp'... not infected
Checking `bindshell'... not infected
Checking `lkm'... not tested
Checking `rexedcs'... not found
Checking `sniffer'... not tested: can't exec ./ifpromisc
Checking `wted'... not tested: can't exec ./chkwtmp
```

The reports from each script, other than fcheck and cops, are appended to that month's total. The filename designates the script name, month, and year that the script generated the report. The log files are saved in /disk2/admin/reports/logs.videoserver.

For example:
login.fails.01.02 (contains all the reports for January 2002 of failed logins)
The fcheck file integrity checker reports are saved in /disk2/admin/reports/fcheck_report.
For example:
02.06.02 (contains the fcheck report for February 6, 2002)
The cops reports are saved in /disk2/admin/cops_104/videoserver.
For example:
2002_Jun_26 (contains the cops report for June 26, 2002)
The reports should be kept for a minimum of one year and then archived to 4mm tape.

B. Quarterly Security Check

1. Run John the Ripper. Execute the following script via cron.

/opt/john-1.6/run/runjohn.sh
The script combines the /etc/passwd and /etc/shadow files. The new file (Testfile) is formatted so John the Ripper can crack the encrypted passwords. A dictionary has been pre-computed to create all permutations of words and removed the duplicates. It is gzipped. We will unzip and feed it the formatted password file Testfile.

```
#!/bin/sh
# Shell script to combine the /etc/passwd and
# /etc/shadow files.
# Then run a dictionary check with rules
# against the newly created password file.
# May 9, 2002
/opt/john-1.6/run/unshadow /etc/passwd /etc/shadow >/opt/john-1.6/run/Testfile
/bin/gzcat /opt/john-1.6/run/MYDICT2.out.gz | /opt/john-1.6/run/john \
- stdin /opt/john-1.6/run/Testfile
```

2. Request a Saint scan. Contact the network security administrator. Make corrections as needed.

3. Remove the Report Files from /disk2/admin/reports directory. Tar them up, save to tape, and store them in the media library. Delete report files. These are just like log files, they grow rapidly.

cd /disk2/admin/reports
Insert a 4mm tape into the 4mm tape drive.
tar cvf /dev/tape *
mt –t /devtape rewind
mt –t /dev/tape offline
mt –t /dev/tape unload
pwd (Make sure you know where you are, you do not want to test your restore procedures unless you have to. However, it is a good habit to get into, to be able to perform a restore and to know your restore tapes are valid).
rm *

C. Content Approval

All training content will be submitted to the training department director. The director will review the material for suitable content and need.
The software quality control group will evaluate any software to be loaded for malicious code, errors, or viruses.

The administrator will encode any video material and install it on the videoserver. The administrator would also install any training software on the vidoserver.

Webserver useage logs should be reviewed to monitor the useage of the applications loaded on the video server. Although logs will be continually monitored, the systems administrator will submit a quarterly report of content useage to the training department director. Content that has not been accessed in the past 6 months will be highlighted. It will be up to the department director to decide if the application is still applicable, needed, or required. The director will reply to the administrator, addressing any concerns or changes that should be made to the video server as a result of the quarterly report. The systems administrator would make the necessary changes.

D. Physical Security

If a user has physical access to the server new problems arise that the operating system cannot protect. A disk can be mounted or stolen. A tape can be used to restore data to any system. It would be easier for someone to steal the backup tapes than it would be to break into the system. After backing up the system, backup tapes should not be left out, where someone could casually walk by and pickup the tapes. A user could shutdown the server and boot to a CD-ROM assuming the PROM password is not set. Sniffers could be attached to the

system. Removing the power cord or the network cable could create a simple denial of service. To combat some of these threats, we will perform the following:

The computer room where the video server is located has a badge access card reader. The department manager and the two administrators are coded to have access to the room. The IRIX 6.5 disks and all update releases of the operating system software are cataloged and kept in the media library. To check media out f the library, the user's badge must be coded to allow access to specific classes of media. All software and backup tapes are cataloged and stored in the media library.

A Halon fire detection system is installed. Although Halon is not environmentally friendly, it is a better choice than a water based system.

An analog phone line and phone is installed incase the phone switchboard goes down in a power outage. A backup power generator is provided to the building in the event of power failure.

V. Testing the Setup of the Server

1. Try to telnet to the videoserver from a Windows 98 client with an address within the intranet.
Log in to a Windows 98 client.
Open a DOS Prompt window.
telnet xxx.xxx.xxx.xxx
A telnet screen opens up with nothing on it. Eventually, the attempt times out. It returns an error message saying: Could not open a connection to xxx.xxx.xxx.xxx
Try to use the hostname to connect.
telnet videoserver
The same thing happened.

2. Try to telnet to the videoserver from a UNIX client with an address within the intranet.
Log in to a UNIX client.
Open a terminal window.
telnet xxx.xxx.xxx.xxx
Telnet: unable to connect to remote host: Connection timed out
Try to use the hostname.
telnet videoserver
The same thing happened.
Tried as user root to telnet to the videoserver from a UNIX client with an address within the intranet.
/sbin/su – (It is a good idea to use the full path to the su executable just in case someone has fooled with root's path statement, to read a trojaned su from some other directory or the current directory.)
entered the root password

telnet xxx.xxx.xxx.xxx
Telnet: unable to connect to remote host: Connection timed out

3. Try to ftp to the videoserver from a Windows 98 client with an address within the intranet.
Log in to a Windows 98 client.
Open a DOS Prompt window.
ftp xxx.xxx.xxx.xxx
FTP: Connect: 10060
ftp>
It did not display any warning or ask for a username and password. I tried ls command. I received the not connected message.
not connected
bye
Try to use the hostname.
ftp videoserver
The same thing happened.

4. Try to ftp to the videoserver from a UNIX client with an address within the intranet.
Log in to a UNIX client.
Open a terminal window.
ftp xxx.xxx.xxx.xxx
FTP: Connect: Connection timed out

On both the telnet and ftp attempts, no information was sent back to the user. Information sent back even when an error occurs can sometimes aid the attacker. Depending on the response certain systems exhibit specific behaviors in a given situation. By observing this response an attacker may be able to identify the operating system or the type of web server. With specific information in hand an attacker can look for specific vulnerabilities against the discovered operating system or web server. It appeared as if the server was down or none existed.

5. Try to ssh to the videoserver from a UNIX client, other than the two allowed hosts.
Logged in to a UNIX client.
Opened a terminal window.
ssh videoserver
connection timed out

ssh xxx.xxx.xxx.xxx
connection timed out

6. Try to ssh to the videoserver from a Windows 98 client running PuTTY other than the two allowed hosts.
Logged in to a Windows 98 client.
Double clicked on the PuTTY icon.
Selected SSH
Entered videoserver in the hostname
Selected Connection
Selected Preferred Protocol 2
Selected Blowfish
Selected Load
Selected Open
Fatal Error Network Connection Timed Out.

7. Try to connect to http://videoserver/mbase from an address outside the intranet subnet.
Logged on to a Windows XP Professional system.
Opened a browser.
Entered the address http://xxx.xxx.xxx.xxx/mbase
HTTP 404 Not Found. Page Cannot be Found.

8. Try to connect to http://videoserver/mbase from the dial in network address. We do not want the employees or anyone else to dial into the network and connect to the video server. This address is blocked by an ACL from the Fasttrack server.
Dialed into the intranet.
Opened a browser.
Entered the address http://videoserver/mbase
HTTP 404 Not Found. Page Cannot be Found.

This was the error that /disk2/admin/scripts/webserver.error.log.sh recorded and was included in the total daily report sent to the root user:

[10/Sep/2002:09:54:50] security: for host xxx.xxx.xxx.xxx trying to GET /mbase/, acl-state reports: access of /var/www/htdocs/mbase/index.cgi denied by ACL default directive 3

9. Try to get a directory listing of the videoserver by entering http://videoserver in the address bar of the browser from an address within the intranet. If the server is misconfigured, a listing of the files will be displayed.
Log in to a Windows 98 client.
Opened a browser.
Entered the address http://videoserver
The page cannot be displayed.

10. Test the backup and restore procedures. Delete some files from the root disk.

Follow these steps to restore the root file system: Retrieve the xfsdump 4mm tape session label root514. (The date is just an example. Full backups and root file system backups should be performed weekly. The backups need to be current so you do not lose any data). Retrieve the latest quarterly update disks (IRIX 6.5.16 disks). Shutdown the system.

shutdown –y –g0
The PROM Menu will be displayed.
Select Install System Software
Select the local CD-ROM
Insert the IRIX 6.5.16 Installation Tools and Overlays [1 of 4] in the CD-ROM drive.
Insert the 4mm xfsdump tape into the 4mm tape drive.
Click on Install
Miniroot is loaded
The inst menu comes up.
Enter shroot
xfsrestore –f /dev/tape /
When it completes enter exit.
type quit
y to Reboot
The system will come back up into multiuser mode. Check to see if the deleted files have been restored.

11. Test the backup and restore procedures. Delete some files from anywhere on the system. We will perform a total system restore. Retrieve the latest full backup tape or tapes. (Depending on how much data is on the system, there may be more than one backup tape). Retrieve the latest quarterly update disks (IRIX 6.5.16 disks). Shutdown the system

shutdown –y –g0
The PROM Menu will be displayed.
Select Recover System
Select the source of recovery.
Select CD-ROM
Insert the IRIX 6.5.16 Installation Tools and Overlays [1 of 4] in the CD-ROM drive.
Insert the 4mm backup tape into the 4mm tape drive.
When the first tape is complete, it will prompt for the next tape or no to stop restoring.
On our system there is just one backup tape.
Answer no
Reboot
The system will come back up into multiuser mode. Check to see if the deleted files have been restored.

12. Contact the network security administrator. Request a Saint scan. The Saint scanner is a network based vulnerability scanner. It does more then just scan ports. It looks for vulnerabilities in services that are running on a server. The results give the administrator on what is going on, on their system. The administrator has more information to make educated decisions on how to configure the server.

Extreme care must be taken when running a vulnerability scanner. A Denial of Service can be created. The network can be slowed down. Depending on how the network is configured, traffic and ports could get shutdown. The system can lock up. It is important to get prior authorization from the Director of security to run a scan. You should notify the system administrators and network administrators of the systems and networks involved, that a scan is going to be run. You do not want to conflict with someone else' work.

The scan report should only be readable as root. If the report is printed out, do not leave them laying out where someone may pick it up or just read it.

The scan triggered many errors trying to get certain pages from the webserver. The webserver.error.log.sh script caught these attempts. It saved it in the report log, it saved it in the webserver error logs, and sent a report to the administrator.

The Saint scan did alert us to the Netscape Fasttrack buffer overflow vulnerability (CVE-1999-0744, CVE-1999-0751, CVE-1999-0752, CVE-1999-0758, CVE-1999-0853). This is a calculated risk to continue to run with a known vulnerability. The risk in minimized by firewall rules preventing addresses outside of our intranet to be dropped. Source port routing is prevented from entering our network. Outgoing connections are monitored and filtered by the network security administrators. Users must have an account on a server on the intranet. There are additional ACL rules in the Fasttrack server restricting access. Ipfiltering is being used. The plan is to upgrade to the Apache server which would eliminate the vulnerability altogether. It is important to keep up with vulnerabilities and patch systems as soon as possible.

There was a caution about snmp. There should not be any exposure here, since the snmp package was removed from the system. The ports involved with snmp have also been blocked. This is an example of analyzing the report to see if there really is a problem. It is not always black and white.

13. Try to write in /usr. This should not work since /usr is mounted ro read only. It would require root to unmount /usr and then mount it as rw (read write).
Log in to a UNIX client as a regular user.
touch /usr/bin/test
touch: /usr/bin/test cannot create

Appendix A - Security References

1. Security Alerts

Sign up for security alerts from the following:
CERT http://www.cert.org Send an email to majordomo@cert.org
SGI http:www.sgi.com sign up for email alerts at
http://www.sgi.com/support/security/wiretap.html
SANS Alert Consensus sans@sans.org
SANS Newsbites sans@sans.org
BUGTRAQ http://www.securityfocus.com/

2. Good Security Web Sites

http://www.sans.org/
http://www.securityportal.com/
http://www.cerias.purdue.edu/coast/
http://www.securityfocus.com
http://www.nsi.org/compsec.html
http://www.CISecurity.org

Appendix B - Procedures

1. Full Restore Procedure

Retrieve the latest full backup tape or tapes. (Depending on how much data is on the system, there may be more than one backup tape). Retrieve the latest quarterly update disks (IRIX 6.5.16 disks). Shutdown the system.

shutdown –y –g0
The PROM Menu will be displayed.
Select Recover System
Select the source of recovery.
Select CD-ROM
Insert the IRIX 6.5.16 Installation Tools and Overlays [1 of 4] in the CD-ROM drive.
Insert the 4mm backup tape into the 4mm tape drive.
When the first tape is complete, it will prompt for the next tape or no to stop restoring.
On our system there is just one backup tape.
Answer no
Reboot
The system will come back up into multiuser mode. Check to see if the deleted files have been restored.

2. Restore the Root Drive

Retrieve the xfsdump 4mm tape session label root514. (The date is just an example. The backups need to be current so you do not lose any data). Retrieve the latest quarterly update disks (IRIX 6.5.16 disks). Shutdown the system.

shutdown –y –g0
The PROM Menu will be displayed.
Select Install System Software
Select the local CD-ROM
Insert the IRIX 6.5.16 Installation Tools and Overlays [1 of 4] in the CD-ROM drive.
Insert the 4mm xfsdump tape into the 4mm tape drive.
Click on Install
Miniroot is loaded
The inst menu comes up.
Enter shroot
xfsrestore –f /dev/tape /
When it completes enter exit.
type quit
y to Reboot
The system will come back up into multiuser mode. Check to see if the deleted files have been restored.

3. Content Approval

All training content will be submitted to the training department director. The director will review the material for suitable content and need.
The software quality control group will evaluate any software to be loaded for malicious code, errors, or viruses.

The administrator will encode any video material and install it on the videoserver. The administrator would also install any training software on the vidoserver.

Webserver useage logs should be reviewed to monitor the useage of the applications loaded on the video server. Although logs will be continually monitored, the systems administrator will submit a quarterly report of content useage to the training department director. Content that has not been accessed in the past 6 months will be highlighted. It will be up to the department director to decide if the application is still applicable, needed, or required. The director will reply to the administrator, addressing any concerns or changes that should be made to the video server as a result of the quarterly report. The systems administrator would make the necessary changes.

Appendix C - List of Security Scripts

1. /disk2/admin/scripts/daily.script.sh

```
#!/bin/sh
```

```
#  This script should be run every night.
#  It calls several other report programs.
#  This script has been modified over the years by
#  numerous admins. There names have been removed
#  to protect their privacy
#
PATH=/usr/bin:/bin
node=`uname -n`
date=`date`
SYSTEM_LOGS=/disk2/admin/reports
SYSTEM_SCRIPTS=/disk2/admin/scripts
cd $SYSTEM_LOGS

exec > $SYSTEM_LOGS/report.stdout
exec 2> $SYSTEM_LOGS/report.stderr

MONTH=`date "+%m"`
YEAR=`date "+%y"`
DATE_NAME=`date "+%m.%d.%y"`
MAIL_DATE=`date "+%Y_%b_%d"`
export MONTH YEAR DATE_NAME MAIL_DATE

# Run the scripts
cd $SYSTEM_SCRIPTS

/bin/sh $SYSTEM_SCRIPTS/su.fails.sh
/bin/sh $SYSTEM_SCRIPTS/su.success.sh
/bin/sh $SYSTEM_SCRIPTS/logins.sh
/bin/sh $SYSTEM_SCRIPTS/login.fails.sh
/bin/sh $SYSTEM_SCRIPTS/find_rhosts.sh
/bin/sh $SYSTEM_SCRIPTS/find_hostsequiv.sh
/bin/sh $SYSTEM_SCRIPTS/find_root_sh_scripts.sh
/bin/sh $SYSTEM_SCRIPTS/auth.log.sh
/bin/sh $SYSTEM_SCRIPTS/webserver.error.log.sh
/bin/sh $SYSTEM_SCRIPTS/md5binchk.sh
/bin/sh $SYSTEM_SCRIPTS/dfout.sh

# run FCHECK file integrity and COPS
/usr/local/fcheck/fcheck -ars | cat >
$SYSTEM_LOGS/fcheck_report/$DATE_NAME
/disk2/admin/cops_104/cops -v

#  Put all the reports into one big report

cd $SYSTEM_LOGS
```

```
cat $SYSTEM_LOGS/su.fails.report.$node \
   $SYSTEM_LOGS/su.success.report.$node \
   $SYSTEM_LOGS/logins.report.$node \
   $SYSTEM_LOGS/login.fails.report.$node \
   $SYSTEM_LOGS/find_hostsequiv.report.$node \
   $SYSTEM_LOGS/find_rhosts.report.$node \
   $SYSTEM_LOGS/find_root_sh_scripts.report.$node \
   $SYSTEM_LOGS/auth.log.report.$node \
   $SYSTEM_LOGS/webserver.errors.report.$node \
   $SYSTEM_LOGS/md5binchk.report.$node.out1 \
   $SYSTEM_LOGS/$node.dfout
                        > $SYSTEM_LOGS/logs.$node/total.report.$node

#  mail total report to root account
cat $SYSTEM_LOGS/logs.$node/total.report.$node | /usr/bin/mail -s "Daily
security report for $node on $date" root@videoserver.trainingcompany.com

cat $SYSTEM_LOGS/fcheck_report/$DATE_NAME | /usr/bin/mail -s "DAILY
FCHECK REPORT for $node on $date" root@videoserver.trainingcompany.com

cat $SYSTEM_LOGS/chkrootkit.report.$node/$DATE_NAME | /usr/bin/mail -s
"DAILY Check Root Kit Report for $node on $date"
root@videoserver.trainingcompany.com

cat /disk2/admin/cops_104/videoserver/$MAIL_DATE | /usr/bin/mail -s "DAILY
COPS REPORT for $node on $date" root@videoserver.trainingcompany.com

/usr/local/fcheck/fcheck -cas
/bin/sh $SYSTEM_SCRIPTS/newlogs.sh
exit 0
```

2. /disk2/admin/scripts/su.fails.sh

```
#!/bin/sh

# Authors have been removed for privacy reasons.
date=`date`
name=`uname -n`
md=`date +%m/%d`
my=`date +%m.%y`
SYSTEM_REPORTS=/disk2/admin/reports

exec > $SYSTEM_REPORTS/su.fails.report.$name
echo "                                        "
echo "-----------------------------------------"
echo "Begin report for failed su attempts for $name on $date"
echo "                                        "
```

```
cat /usr/adm/sulog | grep $md | grep " - "
echo "                                                "
echo "End report for failed su attempts for $name on $date"
echo "------------------------------------------------"
echo "                                                "

# Keep a running log of this report

cat $SYSTEM_REPORTS/su.fails.report.$name >>
$SYSTEM_REPORTS/logs.$name/su.fails.$my

exit 0
```

3. /disk2/admin/scripts/su.success.sh

```
#!/bin/sh

#  Authors removed for privacy reasons.
#

name=`uname -n`
date=`date`
md=`date +%m/%d`
my=`date +%m.%y`
SYSTEM_REPORTS=/disk2/admin/reports

exec > $SYSTEM_REPORTS/su.success.report.$name
echo "                                                "
echo "------------------------------------------------"
echo "Begin report for successful su attempts on $name for $date "
echo "                                                "
cat /usr/adm/sulog | grep $md | grep " + "
echo "                                                "
echo "End report for successful su attempts on $name for $date"
echo "------------------------------------------------"
echo "                                                "
#  Keep a running log of this report
cat $SYSTEM_REPORTS/su.success.report.$name >>
$SYSTEM_REPORTS/logs.$name/su.success.$my

exit 0
```

4. /disk2/admin/scripts/logins.sh

```
#!/bin/sh -v

#  logins.sh
```

```
#  Authors removed for privacy reasons.
#
PATH=/usr/bin:/bin:/etc
ECHO=/bin/echo
DATE=`date`
REPDATE=`date`
node=`uname -n`
my=`date +%m.%y`
echo $my
SYSTEM_REPORTS=/disk2/admin/reports

exec > $SYSTEM_REPORTS/logins.report.$node
exec 2> $SYSTEM_REPORTS/logins.errors.$node

$ECHO ""
$ECHO "-------------------------------------------------------------------"
$ECHO "Begin report of valid logins for $node on $DATE\n"

search=`echo "$DATE" | cut -c1-10`
/usr/bsd/last  | grep "$search"

case "$?" in
  1)    $ECHO "No valid logins to report";;
  2)    $ECHO "Error with grep command in logins.sh";;
  *)    ;;
esac

$ECHO "\nEnd report of valid logins for $node on $DATE"
$ECHO "-------------------------------------------------------------------"

#  Keep a running log of this report

cat $SYSTEM_REPORTS/logins.report.$node >>
$SYSTEM_REPORTS/logs.$node/logins.$my

exit 0
```

5. /disk2/admin/scripts/login.fails.sh

```
#!/bin/sh

#  login.fails.sh
#  Authors removed for privacy reasons.

DATE=`date`
PATH=/usr/bin:/bin
ECHO=/bin/echo
```

```
node=`uname -n`
my=`date +%m.%y`
echo $my
SYSTEM_REPORTS=/disk2/admin/reports

exec > $SYSTEM_REPORTS/login.fails.report.$node
$ECHO ""
$ECHO "------------------------------------------------------------------"
$ECHO "Begin report of failed login attempts (5 conseq) for $node on $DATE\n"
cat /usr/adm/loginlog

# reset loginlog
cp /dev/null /usr/adm/loginlog

case "$?" in
  1)    $ECHO "No failed attempts to report";;
  2)    $ECHO "Error with cat command in login.fails.sh";;
  *)    ;;
esac

$ECHO "\nEnd report of failed login attempts for $node on $DATE"
$ECHO "------------------------------------------------------------------"
#  Keep a running log of this report
cat $SYSTEM_REPORTS/login.fails.report.$node >>
$SYSTEM_REPORTS/logs.$node/login.fails.$my

exit 0
```

6. /disk2/admin/scripts/find_rhosts.sh

```
#!/bin/sh
# Script to find and report any *.rhosts files
# Authors removed for privacy reasons.
#
rm /.rhosts
PATH=/usr/bin:/bin:/etc
ECHO=/bin/echo
DATE=`date`
REPDATE=`date`
node=`uname -n`
my=`date +%m.%y`
SYSTEM_REPORTS=/disk2/admin/reports

exec > $SYSTEM_REPORTS/find_rhosts.report.$node
exec 2> $SYSTEM_REPORTS/find_rhosts.report.$node

$ECHO  ""
```

```
$ECHO  "-------------------------------------------------------------"
$ECHO  "Begin report of find_rhosts for $node on $DATE\n"

find / -name "*.rhosts" -print -exec rm {} \;

$ECHO  "\nEnd report of find_rhosts for $node on $DATE"
$ECHO  "-------------------------------------------------------------"

#  Keep a running log of this report

cat $SYSTEM_REPORTS/find_rhosts.report.$node >>
$SYSTEM_REPORTS/logs.$node/find_rhosts.$my

exit 0
```

7. /disk2/admin/scripts/find_hostsequiv.sh

```
#!/bin/sh
# Script to find /etc/hosts.equiv and remove it.
# Authors removed for privacy reasons.
#
PATH=/usr/bin:/bin:/etc
ECHO=/bin/echo
DATE=`date`
REPDATE=`date`
node=`uname -n`
my=`date +%m.%y`
SYSTEM_REPORTS=/disk2/admin/reports

exec > $SYSTEM_REPORTS/find_hostsequiv.report.$node
exec 2> $SYSTEM_REPORTS/find_hostsequiv.report.$node

$ECHO  ""
$ECHO  "-------------------------------------------------------------"
$ECHO  "Begin report of /etc/hosts.equiv for $node on $DATE\n"

find /etc -name "hosts.equiv" -print -exec rm {} \;

$ECHO  "\nEnd report of /etc/hosts.equiv for $node on $DATE"
$ECHO  "-------------------------------------------------------------"

#  Keep a running log of this report

cat $SYSTEM_REPORTS/find_hostsequiv.report.$node >>
$SYSTEM_REPORTS/logs.$node/find_hostsequiv.$my

exit 0
```

8. /disk2/admin/scripts/find_root_sh_scripts.sh

```
#!/bin/sh
# Script to find and report any setuid root shell scripts
# This script is called by /disk2/admin/scripts/daily.scripts.sh
# Authors removed for privacy reasons.
#
PATH=/usr/bin:/bin:/etc
ECHO=/bin/echo
DATE=`date`
REPDATE=`date`
node=`uname -n`
my=`date +%m.%y`
SYSTEM_REPORTS=/disk2/admin/reports

exec > $SYSTEM_REPORTS/find_root_sh_scripts.report.$node
exec 2> $SYSTEM_REPORTS/find_root_sh_scripts.report.$node

$ECHO  ""
$ECHO  "-------------------------------------------------------------"
$ECHO  "Begin report of find_root_sh_scripts for $node on $DATE\n"

find / -user root -perm 4000 -print

$ECHO  "\nEnd report of find_root_sh_scripts for $node on $DATE"
$ECHO  "-------------------------------------------------------------"

#  Keep a running log of this report

cat $SYSTEM_REPORTS/find_root_sh_scripts.report.$node >>
$SYSTEM_REPORTS/logs.$node/find_root_sh_scripts.$my

exit 0
```

9. /disk2/admin/scripts/auth.log.sh

```
#!/bin/sh

#  Authors removed for privacy reasons.
#  Called by /disk2/admin/scripts/daily.scripts.sh
#  Checks /var/adm/auth.log for that days activity.

name=`uname -n`
date=`date`
md=`date +%m/%d`
my=`date +%m.%y`
#month_day_date="`date +%b\ &d`"
```

```
SYSTEM_REPORTS=/disk2/admin/reports

exec > $SYSTEM_REPORTS/auth.log.report.$name
echo "                                          "
echo "------------------------------------------"
echo "Begin report for auth.log on $name for $date "
echo "                                          "
cat /var/adm/auth.log | grep "`date +%b\ %d`"
echo "                                          "
echo "End report for auth.log on $name for $date"
echo "------------------------------------------"
echo "                                          "
#  Keep a running log of this report
cat $SYSTEM_REPORTS/auth.log.report.$name >>
$SYSTEM_REPORTS/logs.$name/auth.log.$my

exit 0
```

10. /disk2/admin/scripts/webserver.error.log.sh

```
#!/bin/sh

#  Webserver error log
#  It should be executed by the /disk2/admin/scripts/daily.scripts.sh
#  program.

#  Authors removed for privacy reasons.

DATE=`date`
SEARCH_DATE=`date +%d/%h/%Y`
PATH=/usr/bin:/bin
ECHO=/bin/echo
node=`uname -n`
my=`date +%m.%y`
echo $my
SYSTEM_REPORTS=/disk2/admin/reports

exec > $SYSTEM_REPORTS/webserver.errors.report.$node
$ECHO ""
$ECHO "------------------------------------------------------------------"
$ECHO "Begin report of Webserver Errors for $node on $DATE\n"
/usr/bin/cat /var/netscape/fasttrack/httpd-videoserver/logs/errors | grep
"$SEARCH_DATE"

/usr/bin/cat $SYSTEM_REPORTS/webserver.errors.report.$node

$ECHO "\nEnd report of Webserver Errors $node on $DATE"
```

```
$ECHO "-----------------------------------------------------------------"

#  Keep a running log of this report

/usr/bin/cat $SYSTEM_REPORTS/webserver.errors.report.$node >>
$SYSTEM_REPORTS/logs.$node/webserver.errors.$my

exit 0
```

11. /disk2/admin/scripts/md5binchk.sh

```
#!/bin/sh -v

#  md5binchk.sh
#  Calls md5binchk.pl which checks for changes to /usr/bin directory.
#  It creates checksums for selected files. It compares yesterday's
#  and today's. The original is stored offline.
#  Authors removed for privacy reasons.
PATH=/usr/bin:/bin:/etc
ECHO=/bin/echo
DATE=`date`
REPDATE=`date`
node=`uname -n`
my=`date +%m.%y`
echo $my
SYSTEM_REPORTS=/disk2/admin/reports
SYSTEM_SCRIPTS=/disk2/admin/scripts

exec > $SYSTEM_REPORTS/md5binchk.report.$node.out1

$ECHO ""
$ECHO "-----------------------------------------------------------------"
$ECHO "Begin report of MD5 Checksum Check for $node on $DATE\n"
/usr/bin/perl $SYSTEM_SCRIPTS/md5chk.pl
/usr/bin/cat $SYSTEM_REPORTS/md5binchk.report.videoserver.out

$ECHO "\nEnd report of MD5 Checksums for $node on $DATE"
$ECHO "-----------------------------------------------------------------"

#  Keep a running log of this report

cat $SYSTEM_REPORTS/md5binchk.report.$node.out1 >>
$SYSTEM_REPORTS/logs.$node/md5binchk.$my

exit 0
```

12. /disk2/admin/scripts/md5chk.pl

```
#!/usr/bin/perl
rename ("/disk2/admin/reports/md5binchk.report.videoserver",
"/disk2/admin/reports/md5binchk.report.videoserver.old");
@files = ("/usr/bin/chgrp", "/usr/bin/chmod", "/usr/bin/chown", "/usr/bin/cp",
"/usr/bin/csh", "/usr/bin/dd", "/usr/bin/df", "/usr/bin/diff", "/usr/bin/du",
"/usr/bin/find", "/usr/bin/ls", "/usr/bin/mv", "/usr/bin/nm", "/usr/bin/ps", "/usr/bin/rm",
"/usr/bin/script", "/usr/bin/sh", "/usr/bin/su", "/usr/bin/tar", "/usr/bin/who");
foreach $f (@files) {
`/usr/bin/md5sum $f | cat >> /disk2/admin/reports/md5binchk.report.videoserver`;
}
`/usr/bin/diff /disk2/admin/reports/md5binchk.report.videoserver
/disk2/admin/reports/md5binchk.report.videoserver.old | cat >
/disk2/admin/reports/md5binchk.report.videoserver.out`;
```

13. /disk2/admin/scripts/dfout.sh

```
#!/bin/sh
# Purpose:   This script lists all files structures on videoserver
# This script is called by /disk2/admin/scripts/daily.scripts.sh
# Who Should receive a copy = group

group="root"
SYSTEM_LOGS=/disk2/admin/reports
SYSTEM_SCRIPTS=/disk2/admin/scripts
NODE=`uname -n`
exec > $SYSTEM_LOGS/$NODE.dfout
echo ""
echo ""
echo "------------------------------------------------------------------"
echo "   ######################################## "
echo "   FILE STRUCTURES ON `hostname`        "
echo "   ######################################## "
echo ""
echo ""
          # (1) ** df will list all file strucures **
df -k
echo ""
echo ""
echo "    ################################### "
echo "         END OF REPORT              "
echo "------------------------------------------------------------------"
echo ""
echo ""
```

14. /disk2/admin/scripts/chkrootkit.sh

The chkrootkit script is a freeware program that checks for root kits. This is run as part of the daily security script. The package can be downloaded from: http://www.chkrootkit.org/

The following shell script takes the output from the chkrootkit program, formats, and stores the results in a file. Tdaily security script adds the output to the Email to root.

chkrootkit.sh

```
#!/bin/sh
node=`uname -n`
date=`date`
SYSTEM_LOGS=/disk2/admin/reports
SYSTEM_SCRIPTS=/disk2/admin/scripts
cd $SYSTEM_LOGS

exec > $SYSTEM_LOGS/report.stdout
exec 2> $SYSTEM_LOGS/report.stderr


MONTH=`date "+%m"`
YEAR=`date "+%y"`
DATE_NAME=`date "+%m.%d.%y"`
MAIL_DATE=`date "+%Y_%b_%e"`
export MONTH YEAR DATE_NAME MAIL_DATE

cd $SYSTEM_SCRIPTS
$SYSTEM_SCRIPTS/chkrootkit-0.35/chkrootkit | cat >
$SYSTEM_LOGS/chkrootkit.report.$node/$DATE_NAME
exit 0
```

Sources

Pomeranz, Hal, The SANS Institute Soalris Security Step By Step Version 2.0, SANS Institute, 2001.

Pomeranz, Hal, “One-Time Passwords”, Deer Run Associates, 2000.

Pomeranz, Hal “Track 6 Securing UNIX Systems”, SANS Institute Orlando, Fl. April 2002.

Evanoff, Michael, Hardening the IRIX System”, Creative Technology, Inc. http://www.giac.org/cert.php April 17, 2002.

Jones, Keith J., Shema, Mike, & Johnson Bradley C., Anti-Hacker Toolkit, McGraw-Hill / Osborne, 2002.

Johnson, Randy, IRIX System Administration I, SGI Global Education Services, Silicon Graphics, Inc. November 2000.

Burwell, Michael, IRIX System Administration II, SGI Global Education Services, Silicon Graphics, Inc. March 2000.

WSG's SGI IRIX Security Details, http://www.uga.edu/~ucns/wsg/security/sgidetails.html June6, 2002.

Miller, Toby, "Hacker Tools and their Signatures, Part Three: Rootkits", http://online.securityfocus.com/infocus/1228 July10, 2002.

"Securing Your Irix Box", Berkeley Lab Computer Protection Program: Resources, http://www.lbl.gov/ICSD/Security/systems/irix-box-tips.html September 17, 2002.

Berger, Matt, "Virus poses as Microsoft security patch", http://www.nwfusion.com/news/2002/0930msvirus.html October 1, 2002.

Eastwood, Liz and Bernard, Bob, IRIX 6.5 Installation Instructions, Silicon Graphics, Inc. 2002.

IRIX 6.5.16 Update Guide, Silicon Graphics, Inc. May 2002.