Global Information Assurance Certification Paper

Creating a Secure Syslog Server Box under Solaris™ 8 04/01 on a Sun Ultra-10

A Paper Presented for the
GIAC Certification in UNIX Security
SANS Institute
Baltimore, MD May 2001

John Joseph Gerber
July 2001

Contents

List of Tables

Overview

Defining Objective

Before one begins to install Solaris on a box, it is very helpful to define how the box will be used. Knowing what the box will be used for will help define what packages need to be installed and what services will need to run. Knowledge allows for correct choices in the installation process such as network settings (does it need to be networked?), partitions sizes, and packages selections.

The purpose of the box being setup in this paper is to act as a prototype from where the reader may choose and build on what is outlined in the installation procedures. Towards this end, it was decided to go through the procedure of setting up a syslog server. The major functionality of this box is to accept information from other boxes, mainly the system logs. It is a simple server/client process, which, at the very primitive level, can be done using packages existing on the Solaris 8 installation CDs. Before it is connected to the Internet and clients start sending their activity logs to it, additional considerations should be addressed.

By pooling the logs onto one box, the syslog server can see the larger activity picture and possibly detect not only host activities but also network activities. Towards this end, the hosts on the network need to have their system clocks in sync. In our sample network, the NTP server is running off the firewall, which also runs the DNS server. The syslog server needs to be setup as a NTP client.

In order to correctly analyze the log files, the syslog server must have a way to interpret the log files in a timely manner, contacting the administrators quickly when a problem occurs. This is accomplished by attaching a modem to the server and adding paging software. The modem must be configured so it can only make outgoing calls.

Companies have different policies concerning storing of log files. The syslog server will need a method of archiving the files. A tape drive can often suffice using basic UNIX commands to dump data out to tape. In this paper, it is going to be assumed that the requirements are more stringent. Some files need to be placed on media that will not degrade in time and cannot be tampered with. This paper will address this requirement by using a CD-RW driver attached to the server. While the log files are on disk but not being written to, the files can be encrypted for additional security.

With the syslog server acting as a repository for log files, it is very important that the box be tightly secured. Security is about layers. This paper will address some security services that might be useful.

Table 1 outlines the functionality and the software packages that will be used to address the requirements outlined above. This paper will not address the merits of one package

over another. The purpose of this paper is to act as a checklist of what is required to install these packages. The reader must decide what functionality is required. This paper addresses possible services that might be required and outlines how to implement these services in a secure manner. References will be provided for additional information.

Table 1 Required Services

Functionality	**Software**
Syslog Server	Solaris package
NTP client	Solaris package.
Monitoring Logs	Psionic Logcheck
Modem	Solaris configuration and hardware
Paging Software	Q-Page
CD-RW	Solaris 8 packages and hardware
Security Packages	• SunScreen Lite • GNU Privacy Guard • Configurator • TCP Wrapper
Monitoring/Access by System Administrators	SSH

Hardware and Operating System Specifications

This paper addresses the secure setup and installation of software on a particular architecture. It will be left to the reader to adjust the procedures to their architecture. The hardware specifications are:

- Sun Ultra-10
- Two Identical (Manufacturer and Model) 18G Drives
- CD-ROM Internal Drive
- Floppy 1.44Mb Internal Drive
- Plextor PlexWriter 12/10/32S External CD-RW Drive
- 100 Mb Network Connection
- 3Com USRobotics Sportster 56K Modem
- Solaris 8 04/01

The Solaris version and revision levels are important. This paper will take advantage of the latest revision to use recently added commands like "cdrw". The SUNWcdrw was not available until Solaris 8 10/00.

Setup Environment

Physical security of the box is very important. It is assumed that the box will be physically secured. Hal Pomeranz discusses these issues and others in the "Linux/Solaris

Practicum."[1] Another good source to review fundamental security issues is Sun's "System Administration Guide, Volume 2."[2] This document is not meant to duplicate what is well covered in more detail in other guides.

If possible, the box should be setup on a secure network. It can be moved later. This document will outline the installation procedure in a secure manner regardless of the network environment. How the box is connected will affect how software is moved onto the box. During the installation that follows, software was moved onto the box from CDs. Once the box is properly configured, data can be burned onto CDs. Tapes could be used to transfer data. Both methods allow software packages to be loaded from a secure box already setup on the network. Once OpenSSH is installed, secure copy may be used. FTP from outside locations might be used. Attempt to verify the integrity of the software by using checksums or PGP signatures. Do make sure to ftp from community trusted software sites. Even then, there have been occasions where Trojan software has snuck onto trusted sites. The procedure used by the author when ftp is required is to ftp onto an already hardened box. The code is then examined, configured, compiled and when possible made into a package. The packages are then written out onto a toolkit CD. The CD is not only useful for installation but can also be included in incident handling toolkits. Throughout this document, ftp sites will be listed. It is up to the reader to determine what method of data transference can be implemented in their environment.

Initial installation is done with the box not connected to the network. Do not attach the server to the network until specifically told to do so. Inform anyone who has access to the box that it should remain disconnected from the network until further notice.

Development Box

In order to install the production syslog server box with the minimal required software, a second test machine was used. This second machine was the same architecture and operating system. The purpose of the box was to provide a test bed for the examination, configuration, compilation, and installation of software. When possible, the compiled software was made into a package. Later in this paper the idea of using JumpStart for installing packages will be touched upon along with burning in your own JumpStart CD. Having software in package form can be real helpful. In addition, the test box allowed for the examination of programs for the determination of required packages. The test box ran no services outside of SSH, with TCP Wrapper, and was not accessible to the outside world. If setting up such a box is not possible, additional packages will need to be installed, such as GNU C compiler (gcc) and GNU make.

Sample Network

Throughout this document certain boxes will play key roles. The following table can server as a reference in respect to defining host name, IP, and function. Keep this table in mind when examining configuration files, sample results, etc.

Table 2 Sample Configuration for Domain Crobbins.org

Name		IP	Functions
	Kanga	172.16.129.138	• Development/Test Box • Solaris Entire Distribution Installation • On Secure Network
	Roo	172.16.129.178	• Syslog Server • Machine Being Setup
	Pooh	172.16.129.157	• DNS • NTP • Firewall
	Eeyore	172.16.129.11	• Mail Server
	Tigger	172.16.129.136	• Web Server

Conventions Used

Commands and their resulting output will be placed in boxes. The UNIX command prompt appear as a "#". Commands that the user is expected to type in will appear in bold characters. When a file is being edited, the UNIX command to edit will appear at the command line in bold characters, and the file content will appear below in italics characters. The complete path to the command is used. This is to underscore to the reader the need to pay attention to where the commands are located. Users often take their path setting for granted and may find themselves running unexpected commands snuck in by crackers.

Pre-OS Installation

Determining Required Packages

Exploiting security holes in the operating system (OS) is how the majority of system penetrations occur.[3] Reducing the number of unnecessary OS packages installed on the system helps reduce the number of possible vulnerabilities, improving the security of the server. The best practice is to begin with the minimal required and build up.

Solaris™ Installation Clusters

There are four installation clusters available when installing Solaris:

- Core
- End User
- Developer
- Entire Distribution

Kanga, the test machine, has been setup with the Entire Distribution cluster. Kanga will be used to help determine what dependencies exist between packages. Roo will have the Core installation cluster installed in addition to the packages Roo needs. Kanga and documentation will determine Roo needs.

Roo will have 64-bit support. Selecting the Core cluster will result in certain packages being installed. If a 32-bit package was installed, the corresponding 64-bit package was also installed. Appendix A-1 gives a list of the packages that are required by Roo. The legend indicates which packages are included in the Core cluster along with which packages are required for other services. Since the reader will be interested in installing other packages, it is very important to learn the methodology used in determining package requirements. Many system administrators install the entire distribution cluster in order to avoid the hassle of determining package requirements. The following section outlines methods to determine what packages might be required. A complete listing of Solaris 8 packages can be found in the Sun's "Solaris 8 (SPARC Platform Edition) Installation Guide."[4]

Review of Desired Software Packages

Software applications in general do not list required packages, making it difficult to know what packages are required. Determining what packages might be required can be done in stages. During the installation of software, missing required packages might be reported. This is the simplest case.

In a second case, during installation or operations, the software will fail and an error message is generated. The error message will help determine what packages are required.

In the third case, the software might run but operations are degraded or the software core dumps. The Solaris command "truss" can be run to help determine which files were being accessed before the software stopped operating. Going to a machine with an entire distribution cluster installed, like Kanga, allows grep to be used to search for the occurrences of the missing file in the /var/sadm/install/contents file. This can be a time consuming tasks. As a package is installed, the program might advance further but fail on additional missing file.[1]

To address the iterative nature of using the commands truss and grep against the /var/sadm/install/contents file a different approach is possible. Instead of installing the software initially on the installation box (Roo), install the software on the box with the entire distribution cluster installed (Kanga). Start the program under the command "truss" sending the output of truss to a file, perform the required operations, and then stop the program. The truss output will have the listing of files accessed by the program while it was running. The file can be quite large, but fortunately programs written in Perl can be used to parse the files and create a listing of accessed files. The Perl program can then search for occurrences of the accessed files in /var/sadm/install/contents and produce a listing of required packages.

For this paper, the program truss.pl was written to implement the above idea. A file listing appears in Appendix A-3. The program truss.pl is a Perl program that will help determine package dependencies. For clarity, a sample run is provided showing how to determine what packages might be required by NTP. Start by running truss on Kanga, creating a truss output file.

```
# /usr/sbin/truss -f -o /var/tmp/truss.out /etc/rc2.d/S74xntpd start
#
```

After allowing sufficient time for the program to access files, stop the job in another window:

```
# /etc/rc2.d/S74xntpd stop
#
```

If space is a consideration, the output file can be reduced by using the following command:

```
# /usr/sbin/truss -f -o /var/tmp/truss.out -t open -t execve \
      -t start -t access -t creat /etc/rc2.d/S74xntpd start
#
```

The truss.pl program will parse the output file whether or not the "-t" option is used. Depending on how long truss needs to run on the operation and how active the program is will determine if output file size is of concern. Checking the occurrences of execve, start, access, and creat will provide an image of accessed files. This does check both 32-bit and 64-bit operations.

The results of running truss.pl are as follows:

```
# /local/home/jgerber/perl/truss.pl
  SUNWcsd (Core Solaris Devices) required by:
       /dev/conslog
       /dev/null

  SUNWcsl (Core Solaris, (Shared Libs)) required by:
       /usr/lib/libaio.so.1
       /usr/lib/libc.so.1
       /usr/lib/libdl.so.1
       /usr/lib/libelf.so.1
       /usr/lib/libl.so.1
       /usr/lib/libmd5.so.1
       /usr/lib/libmp.so.2
       /usr/lib/libnsl.so.1
       /usr/lib/librt.so.1
       /usr/lib/libsocket.so.1

  SUNWcsr (Core Solaris, (Root)) required by:
       /sbin/sh

  SUNWcsu (Core Solaris, (Usr)) required by:
       /usr/bin/cat
       /usr/bin/sleep

  SUNWesu (Extended System Utilities) required by:
       /usr/bin/nawk

  SUNWlibms (Sun WorkShop Bundled shared libm) required by:
       /usr/lib/libm.so.1

  SUNWntpr (NTP, (Root)) required by:
       /etc/rc2.d/S74xntpd

  SUNWntpu (NTP, (Usr)) required by:
       /usr/lib/inet/xntpd
       /usr/sbin/ntpdate

  SUNWscpu (Source Compatibility, (Usr)) required by:
       /usr/share/lib/zoneinfo/US/Eastern
#
```

Non-volatile RAM (NVRAM) Variables

NVRAM variables store system configuration variables that affect a variety of operations. There are a few NVRAM variables under the SPARC architecture that should be set to improve console security.[5] These NVRAM security variables are:

- `security-mode`
- `security-password`
- `security-#badlogins`
- `oem-banner?`
- `oem-banner`
- `local-mac-address`

A complete listing of NVRAM variables and functions exist in Sun's document "OpenBoot 3.x Command Reference Manual."[6] This document will concentrate on just the above variables.

There are three possible settings for security-mode and these values are described in Table 3.

Table 3 Values for security-mode

Value	Commands
`Full`	All commands except for `go` require the password.
`Command`	All commands except for `boot` and `go` require the password.
`None`	No password required (default).

To restrict the operations that can be performed from the console, set security-mode to the most restrictive value of full. Setting security-mode to full will result in:

- A password being required any time the boot command is issued.
- The go command will not ask for a password (true with all settings).
- A password is required to execute any other command.

Setting security-mode at the OpenBoot PROM is done as follows:

```
ok password
ok New password (only first 8 chars are used):
ok Retype new password:
ok setenv security-mode full
ok
```

The security-mode and security-password can be set from the UNIX prompt as follows:

```
# eeprom security-mode=full
  New password (only first 8 chars are used):
  Retype new password:
#
```

For the rest of this section, it will be assumed that the NVRAM variables are being set from the 'ok' prompt (OpenBoot PROM). To get to the 'ok' prompt press the Stop and A key (Stop-A) at the same time.

The security-#badlogins is not a variable that is set but rather it provides information on how many bad login attempts have been made from the OpenBoot PROM. To view this information:

```
ok security-#badlogins
   security-#badlogins=3
ok
```

The counter can be reset with the command:

```
ok security-#badlogins=0
   security-#badlogins=3\0
ok
```

The banner configuration variables should be set as follows:

```
ok setenv oem-banner Authorized users only.  All access may be logged and
reported.
ok setenv oem-banner? true
ok banner
   Authorized users only.  All access may be logged and reported.
ok
```

While the machine, Roo, is not a multi-homed machine, consider setting the local-mac-address at this time in order to avoid having to set it in the future. Setting will have no negative affect on the current configuration.

```
ok setenv local-mac-address? true
ok
```

Installation Solaris 8 Update 04/01

This section will provide the reader with a checklist of values to enter while progressing through the interactive installation of the Solaris 8 update 04/01. This installation is for the domain described in Table 2. Roo is being installed with an initial 64-bit installation. For a complete description of the installation process, the reader should consult Sun's document "Solaris 8 Advance Installation Guide."[7]

Booting the System

At the ok prompt, boot from the CD-ROM drive.

```
ok boot cdrom
ok
```

Information similar to the following will be displayed:

```
Boot device: /pci@1f,0/pci@1,1/ide@3/cdrom@2,0:f  File and args:
SunOS Release 5.8 Version Generic_108528-07 64-bit
Copyright 1983-2001 Sun Microsystems, Inc.  All rights reserved.
Configuring /dev and /devices
Using RPC Bootparams for network configuration information.
SUNW,hme0: No response from Ethernet network " Link down - cable
problem?
```

Text Based Installation Section

For the text based installation section, refer to Table 4 for the values that should be entered at the indicated prompts.

Table 4 Response to Text Based Installation Section

Prompt	Selection
Select a Language	0 English
Select a Locale	0. English (C – 7-bit ASCSII)

OpenWindows™ Desktop Installation Section

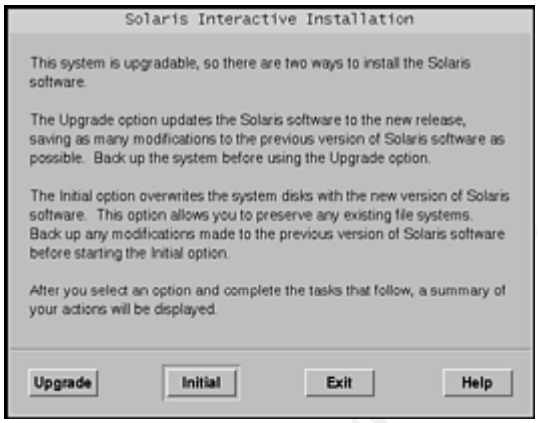
The OpenWindows ™ desktop installation phase will now begin. This section can be divided into two parts. Table 5 provides the values for the identification of the system.

Table 5 Response to Identifying the System Section

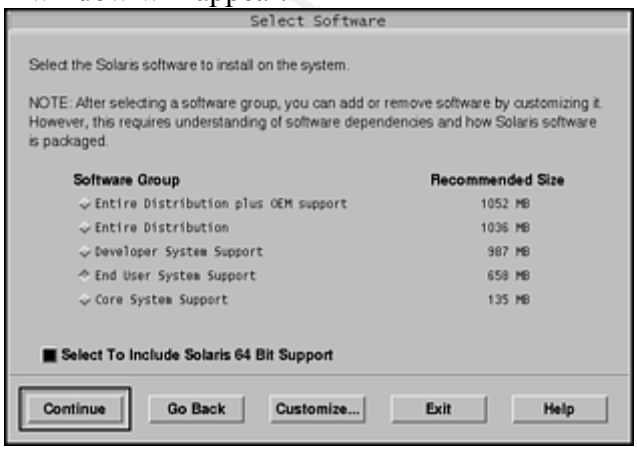
Window Header	Selection
Hostname	Roo
Network Connectivity	Yes
DHCP	No
IP Address	172.16.129.178

Subnets	Yes
Netmask	255.255.255.0
IPv6	No
Configure Security Policy	No
Name Service	DNS
Domain Name	Crobbins.org
DNS Server Address	172.16.129.157
DNS Search List	Continue
Time Zone	Geographic region
Region	Eastern

Depending whether the system had anything previously installed, a window will be displayed similar to:



Select the "Initial" button. The system will prompt for a geographic region. After pressing the continue button in the "Select a Geographic Region" window, the software selection window will appear.



Select "Core System Support" and have the "Select To Include Solaris 64 But Support" box checked. After making this selection, press the "Customize" button. Appendix A-1 gives a detail list of packages that should be selected. Appendix A-1 lists the packages in the groupings, order, and with descriptions that the install process uses. This should make finding the packages easier. After selecting all the packages, the select software

window will appear again. At this point select the "Continue" button. Table 6 provides the responses to the software installation prompts that will follow.

Table 6 Response to Installing the Solaris 8 Software

Window Header	Selection
Select a Geographic Region	Continue
Select Software	• Core System Support • Customize • Select Packages • Continue
Select Disks	• c0t0d0 • Continue
Automatically Layout File System?	Auto Layout
Preserve Data?	Continue
Automatically Layout File Systems	Select • / • /opt • /usr • /var • /swap Continue
File System and Disk Layout	Customize

After selecting the "Customize" button for file layout, values for the partitioning will need to be selected. Roo configuration is described in Table 7. This layout is based on the layout discussed on page 115 of the "6.5 Linux/Solaris Practicum."[1]

Table 7 Customized Disk Layout

Slice	Partition	Size
0	/	256 MB
1	/var	2048 MB
2	Backup	
3	Swap	2048 MB
4	/usr	1024 MB
5	/opt	1024MB
6	/local	Remaining
7	/vcd	2048 MB

The partition /vcd will be a working area for the creation of CDs. After selecting the drive layout, when prompted if the system should mount remote file systems, press the "Continue" button. A profile window will appear with the information entered in this section. To begin installation, press the "Begin Installation" button. Press the "Auto reboot" button when asked if the system should do an auto or manual reboot. At this point the Solaris 8 operating environment will set up the partitions and install the

packages. The installation log file, after the reboot, is /var/sadm/system/logs/install_logs.

Post OS Installation (Pre-Network)

Removing Packages

The Solaris 8 Core cluster comes with several packages marked as being required that might not be desirable on the system. In Alex Noordergraaf paper, "Solaris™ Operating Environment Minimization for Secuity"[3], for a sun4u/SPAC/PCI/headless system he was able to remove half of the packages included in the Core Cluster. In Appendix A-3, the packages marked with an "*" will be removed at this point.

```
# for i in SUNWatfsr SUNWftpr SUNWm64 SUNWm64x SUNWnisr SUNWpcelx \
SUNWpsdpr SUNWpcmci SUNWpcmem SUNWpcser SUNWses SUNWsesx SUNWssad \
SUNWssadx SUNWsolnm SUNWluxd SUNWluxdx SUNWfcip SUNWfcp SUNWfctl \
SUNWfcipx SUNWfcpx SUNWfctlx SUNWauda SUNWaudd SUNWauddx SUNWatfsu \
SUNWftpu SUNWnisu SUNWpcmcu SUNWrmodu SUNWluxop SUNWluxox SUNWadmr \
SUNWpcmcx SUNWpl5u SUNWwsr2 SUNWusb SUNWusbx
> do
> /usr/sbin/pkgrm ${i}
> done
#
```

Additional Packages

As of Solaris 8 10/00, Sun had made available a package, SUNWcdrw. This package provides the ability to create data and audio CDs.

Place "Software Supplemental for the Solaris™ 8 Operating Environment" CD into the CD-ROM drive and issue the commands needed to install the software.

```
# /usr/sbin/mount -F hsfs -o ro /dev/dsk/c0t2d0s0 /cdrom
# cd /cdom/CDRW_1.0/Product
# /usr/sbin/pkgadd -d . SUNWcdrw
#
```

In order to get a listing of the CD devices connected to the system, use the "-l" option as follows:

```
# cdrw -l
#
```

A response will appear similar to:

Looking for CD devices…

Node	Connected Devices			Device type
/dev/rdsk/c0t2d0s2	LG	CD-ROM	CRD-8483B 1.00	CD Reader
/dev/rdsk/c1t4d0s2	PLEXTOR	CD-R	PX-W1210S 1.01	CD Reader/Writer

While the command cdrw will operate whether or not volmgt is running, the output for the above command is affected. The output above was generated with volmgt stopped. The second package required for useful CD operations is SUNWmkcd. This package was installed with the OS.

Disabling Services

Before connecting the machine to the network, disable the startup commands that will result in open ports on the system. A more complete disabling of services will be done in the next section. Right now, the objective is to get the box ready for the network in order for us to start getting security software packages. Even if the service is to run later with TCP Wrapper, initially it needs to be disabled. If packages are being transferred using tapes or CDs, connection to the network can be delayed.

```
# cd /etc/rc2.d
# for i in S72inetsvc S69inet S71ldap.client S71rpc S71sysid.sys
S72autoinstall \
S73cachefs.daemon S73nfs.client S76nscd S80PRESERVE S88sendmail S88utmpd \
93cacheos.finish
> do
> /usr/bin/mv ${i} .no${i}
> done
#
```

Connecting Network

After connecting to the network, running the program nmap from Kanga will show no open port off Roo:

```
# nmap 172.16.129.178
Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/
)
Interesting ports on  (172.16.129.178):
(The 1522 ports scanned but not shown below are in state: closed)
Port       State       Service
#
```

Hardening Solaris

There are many fine papers and guides on securing Solaris. Hal Pomeranz course book[1] or the more concise guide from SANS, "Solaris Security Step by Step"[8] are very good references. One of the quickest ways to secure a box is to install a hardening tool. Make sure to understand what the tool does before running the software. By using Hal Pomeranz "Configurator Toolkit", some important but very well covered topics can be bypassed. The reader should refer to the references above for a better understanding of these security essentials. The README and INSTALL documents and software for the configurator toolkit are provided off the Deer Run Associates web site.[9] The software package provides additional documentation the reader should examine.

This section covers how to configure and install OpenSSH with supporting security software. The configurator toolkit will be used to tighten security and install patches. Sun's Automated Security Enhancement Tool (ASET) will be used to tighten system security further and generate security logs. Auditing files are modified. System accounting is activated. Network access will be restricted further by installing the SunScreen Lite firewall.

Please note that GNU C compiler (gcc) and GNU make will need to be installed if compilation is to be done on Roo. All are available as packages off http://www.sunfreeware.com. As discussed previously, in this paper the compilation is done off Kanga and the finished compiled package is moved to Roo.

Encryption and Checksums

In order to ensure that the packages coming from sites have not been altered, install software that will help check data integrity. Most sites will provide MD5 checksums or PGP signatures for packages. The PGP software can be configured to use a module for the use of random byte gathering. For this purpose, PRNGD will be used. PRNGD will generate random numbers used in cryptographic operations, providing a constantly filled entropy pool. The following section outlines how to install the package MD5 and how to configure and install both PRNGD and GNU Privacy Guard.

1. Pull down the md5 package from:

   http://www.sunfreeware.com/programlistsparc8#md5

2. Before installing any packages, create the directory /opt/local and make /usr/local a link to it. Some packages, will want to install in /usr/local, while other packages will install in /opt/local. This will avoid path confusion.

```
# mkdir -p -m 755 /opt/local
# ln -s /opt/local /usr/local
#
```

3. Uncompress, install the package, and use it to check it's own signature.

```
# cd /local/software
# /usr/sbin/pkgadd -d  md5-6142000-sol8-sparc-local

The following packages are available:
  1  SMCmd5     md5
                (sparc) 6142000

Select package(s) you wish to process (or 'all' to process
all packages). (default: all) [?,??,q]: 1
#
```

4. Download the PRNGD software from:

   http://www.aet.tu-cottbus.de/personen/jaenicke/postfix_tls/prngd.html

5. Check the MD5 checksum, configure, compile, and install PRNGD.

```
# cd /local/software
# /usr/local/bin/md5 prngd-0.9.19.tar.gz
MD5 (prngd-0.9.19.tar.gz) = d17dff7bb69dca79cbf44fdd5389e912
# /usr/bin/gunzip prngd-0.9.19.tar.gz
# /usr/bin/tar xf prngd-0.9.19.tar
# cd prngd-0.9.19
# /usr/ccs/bin/make CC=gcc CGLAGS="-03 -DSOLARIS" SYSLIBS="-lsocket -
lnsl"
# /usr/bin/cp prngd /usr/local/sbin/prngd
# /usr/bin/chown root:bin /usr/local/sbin/prngd
# /usr/bin/chmod 755 /usr/local/sbin/prngd
# /usr/bin/cp contrib/Solaris-7/prngd.conf.solaris-7 /etc/prngd.conf
# /usr/bin/cat /var/log/syslog > /etc/prngd-seed
# /usr/bin/vi /etc/init.d/prngd
#!/sbin/sh
#

case "$1" in
'start')
  if [ -f /usr/local/sbin/prngd ]; then
     echo "Starting PRNGD."
     /usr/local/sbin/prngd /var/run/egd-pool > /dev/null
  fi
  ;;
'stop')
  if [ -f /usr/local/sbin/prngd ]; then
    echo "Stopping PRNGD."
    pgpid=`/usr/bin/ps -e -o fname,pid | /usr/bin/grep prngd | \
       /usr/bin/grep -v grep | /usr/bin/awk '{print $2}'`
    if [ "x$pgpid" != "x" ]; then
      kill $pgpid > /dev/null 2>&1
    fi
  fi
  ;;
*)
  echo "Usage: prngd { start | stop }"
  ;;
esac
exit 0
# /usr/bin/chmod u+x /etc/init.d/prngd
# /usr/bin/ln -s /etc/init.d/prngd /etc/rc3.d/S99prngd
# /etc/rc3.d/S99prngd start
#
```

6. Pull down the GNU dbm (gdbm) package. It is used by GNU Privacy Guard.

   http://www.sunfreeware.com/programlistsparc8.html#gdbm

7. Install the GNU dbm package.

```
# cd /local/software
# /usr/sbin/pkgadd -d gdbm-1.8.0-sol8-sparc-local
The following packages are available:
  1  SMCgdbm     gdbm
                 (sparc) 1.8.0

Select package(s) you wish to process (or 'all' to process
  all packages). (default: all) [?,??,q]: 1
#
```

8. Download GNU Privacy Guard from:

   http://www.gnupg.org/download.html

9. Check the MD5 checksum.

```
# cd /local/software
# /usr/local/bin/md5 gnupg-1.0.6.tar.gz
MD5 (gnupg-1.0.6.tar.gz) = 7c319a9e5e70ad9bc3bf0d7b5008a508
#
```

10. Configure, compile, and install GNU Privacy Guard.

```
# cd /local/software
# /usr/bin/gunzip gnupg-1.0.6.tar.gz
# /usr/bin/tar xf gnupg-1.0.6.tar
# cd gnupg-1.0.6
# ./configure --enable-static-rnd=egd --with-egd-socket=/var/run/egd-
pool
# /usr/local/bin/make
# /usr/local/bin/make check
# /usr/local/bin/make install
#
```

11. Generate a new key-pair.

```
# /usr/local/bin/gpg --gen-key
gpg (GnuPG) 1.0.6; Copyright (C) 2001 Free Software Foundation, Inc.
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions. See the file COPYING for details.

gpg: /local/home/jgerber/.gnupg: directory created
gpg: /local/home/jgerber/.gnupg/options: new options file created
gpg: you have to start GnuPG again, so it can read the new options file
# /usr/local/bin/gpg --gen-key
gpg (GnuPG) 1.0.6; Copyright (C) 2001 Free Software Foundation, Inc.
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions. See the file COPYING for details.

gpg: /local/home/jgerber/.gnupg/secring.gpg: keyring created
gpg: /local/home/jgerber/.gnupg/pubring.gpg: keyring created
Please select what kind of key you want:
   (1) DSA and ElGamal (default)
   (2) DSA (sign only)
   (4) ElGamal (sign and encrypt)
Your selection? 1
DSA keypair will have 1024 bits.
About to generate a new ELG-E keypair.
              minimum keysize is  768 bits
              default keysize is 1024 bits
    highest suggested keysize is 2048 bits
What keysize do you want? (1024) 1024
Requested keysize is 1024 bits
Please specify how long the key should be valid.
         0 = key does not expire
      <n>  = key expires in n days
      <n>w = key expires in n weeks
      <n>m = key expires in n months
      <n>y = key expires in n years
Key is valid for? (0) 30
Key expires at Mon Aug 20 11:12:58 2001 EDT
Is this correct (y/n)? y

You need a User-ID to identify your key; the software constructs the
user id
from Real Name, Comment and Email Address in this form:
    "Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"

Real name: John Gerber
Email address: John.J.Gerber@crobbins.org
Comment:
You selected this USER-ID:
    "John Gerber <John.J.Gerber@crobbins.org>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? O
You need a Passphrase to protect your secret key.

We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
++++++++++++++++++++++++++++++....++++++++++.+++++..++++++++++++++++++
.
++++++++++.++++++++++++++++++++++++++++++++....+++++.++++++++++.+++++
>
..++++++++++>+++++........<+++++.................+++++
Please wait, entropy is being gathered. Do some work if it would
keep you from getting bored, because it will improve the quality
of the entropy.
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
+++++.++++++++++++++++++++++++++++++++++++++++++..++++++++++++++++++
+
.++++++++++..++++++++++.++++++++++.++++++++++++++++++++++++.+++++>+++++
>
+++++.................................................+++++^^^
public and secret key created and signed.
#
```

OpenSSH

OpenSSH allows for encrypted connections between hosts. Jason Reid and Keith Watson have written a paper that provides additional information on installing OpenSSH entitled, "Building and Deploying OpenSSH for the Solaris™ Operating Environment."[10] A few software packages need to be installed prior to OpenSSH. Table 8 provides a listing of the software and where it can be downloaded. Zlib is a library that contains the data compression algorithms used by OpenSSH. OpenSSL contains cryptographic algorithms and can be compiled without PRNGD, but the results binaries will pause while the OpenSSH algorithms perform entropy collections. PRNGD provides a constantly filled entropy pool. TCP Wrapper allows OpenSSH to restrict hosts by IP address.

Table 8 Creating an OpenSSH Package

Software Package	URL
OpenSSH	http://www.openssh.com/portable.html
Zlib	Already installed off Solaris OS CDs.
OpenSSL	http://www.openssl.org/source
PRNGD	Already installed in last section.
TCP Wrapper	ftp://ftp.porcupine.org/pub/security/index.html

1. Get TCP Wrapper, the signature file, and Wietse Zweitze's PGP public key from:

   ftp://ftp.porcupine.org/pub/security/

2. There is no guarantee that the PGP public key is valid. The topic of trust is outside the scope of this paper. The point is to demonstrate how to check PGP signatures. Import Wietse's PGP public key and use the signature file to verify software. Gunzip, untar, compile, and install TCP Wrapper[8].

```
# cd /local/software
# /usr/local/bin/gpg --import wietse.pgp
gpg: key D5327CB9: public key imported
gpg: /local/home/jgerber/.gnupg/trustdb.gpg: trustdb created
gpg: Total number processed: 1
gpg:               imported: 1  (RSA: 1)
# /usr/local/bin/gpg --verify tcp_wrappers_7.6.tar.gz.sig \
   tcp_wrappers_7.6.tar.gz
gpg: Signature made Mon Apr 07 20:43:23 1997 EDT using RSA key ID
D5327CB9
gpg: Good signature from "wietse venema <wietse@porcupine.org>"
gpg:                 aka "wietse venema <wietse@wzv.win.tue.nl>"
Could not find a valid trust path to the key.  Let's see whether we
can assign some missing owner trust values.

No path leading to one of our keys found.

gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the
owner.
gpg: Fingerprint: 78 96 4A 4D F0 F0 D1 3C  45 E9 03 FC 17 67 DC D8
# /usr/bin/gunzip -c tcp_wrappers_7.6.tar.gz | tar xf -
# cd tcp_wrappers_7.6
# make CC=gcc REAL_DAEMON_DIR=/usr/sbin FACILITY=LOG_AUTH sunos5
# mkdir -m 755 -p /usr/local/sbin /usr/local/include /usr/local/lib
# for i in safe_finger tcpd tcpdchk tcpdmatch try-from
> do
> /usr/sbin/install -s -f /usr/local/sbin -m 555 -u root -g daemon $i
> done
# /usr/sbin/install -s -f /usr/local/include -m 444 -u root -g daemon
tcpd.h
# /usr/sbin/install -s -f /usr/local/lib -m 555 -u root -g daemon
libwrap.a
#
```

3. Pull down the OpenSSL software, check the MD5 checksum, configure, and compile the software.

```
# cd /local/software
# /usr/local/bin/md5 openssl-0.9.6b.tar.gz
MD5 (openssl-0.9.6b.tar.gz) = bd8c4d8c5bafc7a4d55d152989fdb327
# /usr/bin/gunzip -c openssl-0.9.6b.tar.gz  | /usr/bin/tar xf -
# cd openssl-0.9.6b
# /bin/sh ./config
# /usr/local/bin/make
# /usr/local/bin/make test
# /usr/local/bin/make install
#
```

4. Pull down the OpenSSH software, the OpenSSH PGP key, and the signaturefile. Import the public key, check the software against the signature, configure, make, and install.

```
# cd /local/software
# /usr/local/bin/gpg --import DJM-GPG-KEY.asc
gpg: key 86FF9C48: public key imported
gpg: Total number processed: 1
gpg:               imported: 1
# /usr/local/bin/gpg --verify openssh-2.9p2.tar.gz.sig \
  openssh-2.9p2.tar.gz
gpg: Signature made Sun Jun 17 00:20:30 2001 EDT using DSA key ID
86FF9C48
gpg: Good signature from "Damien Miller (Personal Key) <djm@mindrot.org>"
Could not find a valid trust path to the key.  Let's see whether we
can assign some missing owner trust values.

No path leading to one of our keys found.

gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the
owner.
gpg: Fingerprint: 3981 992A 1523 ABA0 79DB  FC66 CE8E CB03 86FF 9C48
# /usr/bin/gunzip -c openssh-2.9p2.tar.gz | /usr/bin/tar xf -
# cd openssh-2.9p2
# CFLAGS="-I/usr/local/include" LDFLAGS="-L/usr/local/lib" ./configure \
  --prefix=/usr/local --with-tcp-wrappers --without-rsh --disable-suid-
ssh \
  --sysconfdir=/etc --with-prngd-socket=/var/run/egd-pool
# /usr/local/bin/make
# /usr/local/bin/make install
#
```

5. Edit the /etc/hosts.allow, /etc/hosts.deny, and /etc/sshd_config configuration files. The /etc/sshd_config can be copied from the OpenSSH software directory and then modified.

```
# cd /local/software/paper/openssh-2.9p2
# /usr/bin/cp sshd_config /etc/sshd_config
# /usr/bin/vi /etc/sshd_config
Port 22
Protocol 2,1
ListenAddress 0.0.0.0
HostKey /etc/ssh_host_key
HostKey /etc/ssh_host_rsa_key
HostKey /etc/ssh_host_dsa_key
ServerKeyBits 1024
LoginGraceTime 180
KeyRegenerationInterval 900
PermitRootLogin no
IgnoreRhosts yes
IgnoreUserKnownHosts yes
StrictModes yes
X11Forwarding yes
X11DisplayOffset 10
PrintMotd no
KeepAlive no
SyslogFacility AUTH
LogLevel INFO
RhostsAuthentication no
RhostsRSAAuthentication no
HostbasedAuthentication no
RSAAuthentication yes
PasswordAuthentication yes
PermitEmptyPasswords no
CheckMail no
UseLogin no

# /usr/bin/vi /etc/hosts.allow
# Allow hosts access to sshd port
sshd: 172.16.129.138
# /usr/bin/vi /etc/hosts.deny
ALL: ALL: /usr/bin/mailx -s "%s: connection attempt from %a" sysadmin-
page
#
```

6. Create the startup script for OpenSSH.

```
# /usr/bin/vi /etc/init.d/sshd
#!/sbin/sh
#

case "$1" in
'start')
  if [ -x /usr/local/sbin/sshd -a -f /etc/sshd_config ]; then
     echo "Starting SSHD."
     /usr/local/sbin/sshd -f /etc/sshd_config
  fi
  ;;
'stop')
  kill `/usr/bin/cat /etc/sshd.pid`
  ;;
*)
  echo "Usage: sshd { start | stop }"
  ;;
esac
exit 0
```

Configurator Toolkit

The configurator toolkit provides the functionality of not only being able to check the system and harden it by making modifications, but it will also install patch clusters, packages, and copy software into place. The configurator toolkit will be used to install the latest Sun patch cluster.

1. Get the most recent Solaris 8 patch cluster from:

   http://sunsolve.Sun.COM/pub-cgi/show.pl?target=patches/patch-access

2. Check the MD5 checksum.

```
# cd /local/software
# /usr/local/bin/md5 8_Recommended.zip
MD5 (8_Recommended.zip) = efbb6713e3902a3c2e9d744c6c35beae
#
```

3. The configurator toolkit is available from:

   http://www.deer-run.com/~hal/jumpstart/configurator/

4. Unpack the software.

```
# cd /local/software
# /usr/bin/uncompress config.tar.Z
# /usr/bin/tar xf config.tar
# cd /local/software/configurator
#
```

5. Move the Sun recommended patches into the configurator/patches directory. Configure configurator. Modify the bin/configurator file so CONFROOT has the value of "/local/software/configurator." Make sure to modify the variable DEFAULTROUTER so it has a value of "172.16.129.157". Run the configurator toolkit.

```
# cd /local/software
# /usr/bin/mv 8_Recommended.zip  configurator/patches
# cd configurator/patches
# /usr/bin/unzip -qq 8_Recommended.zip
# /usr/bin/mv 8_Recommended 5.8-sparc
# /usr/bin/rm 8_Recommended.*
# cd ../conf
# /usr/bin/vi Defaults
DEFAULTROUTER='198.125.128.157'
# cd ../bin
# /usr/bin/vi configurator
CONFROOT=/local/software/configurator
# ./configurator
Picking up settings: Delfaults  done.
completed patchadd
 cd /local/software/paper/configurator/patches/5.8-sparc/install_cluster
 cd /local/software/paper/configurator/patches/5.8-sparc

Patch cluster install script for Solaris 8 Recommended

The nosave option was used.  Objects will not be saved.
Installing patches located in /local/software/paper/configurator/patches/5.8-sparc
Using patch_order file for patch installation sequence
Installing 108725-05...
Etc…
For more installation messages refer to the installation logfile:
  /var/sadm/install_data/Solaris_8_Recommended_log

Use '/usr/bin/showrev -p' to verify installed patch-ids.
Refer to individual patch README files for more patch detail.
Rebooting the system is usually necessary after installation.
Installing local configuration files:tar: .[A-Za-z0-9]*: No such file or directory
tar: *: No such file or directory
 done.
Running scripts: set_root_password update_nsswitch update_vfstab purge_boot_seq
 new_devfsadm  create_umask_scr new_netconfigln new_inetsvc update_inittab
 purge_files create_defaultrouter sendmail_via_cron purge_users create_ftpusers
 update_syslog enable_accounting create_warnings update_etc_defaults
 update_rmmount.conf update_etc_system disable_rhosts update_cron_allow enable_bsm
 set_crontab_perms run_fix-modes done.
#
```

6. Reboot the system for the patches and configurations changes to be put in place..

```
# /usr/sbin/reboot -r
#
```

Automated Security Enhancement Tools (ASET)

ASET allow the administrator to set a security level. The package tightens file permissions, checks system files, and monitors crucial areas. Additional information is available from Sun's "System Administration Guide, Volume 2."[2]

1. ASET is installed under /usr/aset and writes to directories under /usr/aset. The /usr partition is to be mounted read only. Move the directories that ASET uses for writing to another partition. Configure ASET, if necessary. ASET uses the /usr/aset/asetenv file and the /usr/asset/masters directory to define how it operates. The defaults are good for initial operations

```
# /usr/bin/mkdir -p -m 700 /local/security/aset
# cd /usr/aset
# for i in archives tmp reports
> do
> /usr/bin/find ${i} -print | /usr/bin/cpio -pudm
/local/security/aset
> /usr/bin/rm -r ${i}
> /usr/bin/ln -s /local/security/aset/${i}
> done
0 blocks
0 blocks
0 blocks
# /usr/aset/aset -l high
======= ASET Execution Log =======

ASET running at security level high

Machine = roo; Current time = 0722_12:19

aset: Using /usr/aset as working directory

Executing task list ...
        env
        sysconf
        usrgrp
        tune
        cklist
        eeprom
All tasks executed. Some background tasks may still be running.

Run /usr/aset/util/taskstat to check their status:
     /usr/aset/util/taskstat     [aset_dir]

where aset_dir is ASET's operating directory, currently=/usr/aset.

When the tasks complete, the reports can be found in:
     /usr/aset/reports/latest/*.rpt
You can view them by:
     more /usr/aset/reports/latest/*.rpt
#
```

2. Examine the reports.

```
# /usr/bin/more /usr/aset/reports/0723_03:31/*rpt
::::::::::::::
/usr/aset/reports/0723_03:31/cklist.rpt
::::::::::::::

*** Begin Checklist Task ***

... Checklist snapshot is being created. Wait ...
... Checklist snapshot created.

Here are the differences in the checklist.
…
<skipping>
..
*** End Checklist Task ***
::::::::::::::
/usr/aset/reports/0723_03:31/eeprom.rpt
::::::::::::::

*** Begin EEPROM Check ***

EEPROM security option currently set to "full".
::::::::::::::
/usr/aset/reports/0723_03:31/env.rpt
::::::::::::::

*** Begin Enviroment Check ***

*** End Enviroment Check ***
::::::::::::::
/usr/aset/reports/0723_03:31/firewall.rpt
::::::::::::::

*** Begin Firewall Task ***

IP forwarding already disabled.

IP forwarding already disabled in rc files.

ROUTED daemon already configured to be opaque.

*** End Firewall Task ***
::::::::::::::
/usr/aset/reports/0723_03:31/sysconf.rpt
::::::::::::::

*** Begin System Scripts Check ***
chmod: WARNING: can't access /var/adm/utmp
chmod: WARNING: can't access /var/adm/wtmp

World writability for /var/adm/utmp & /var/adm/utmpx has been
removed.
World writability for /var/adm/wtmp & /var/adm/wtmpx has been
removed.

*** End System Scripts Check ***
::::::::::::::
/usr/aset/reports/0723_03:31/tune.rpt
::::::::::::::

*** Begin Tune Task ***

... setting attributes on the system objects defined in
    /usr/aset/masters/tune.high

*** End Tune Task ***
::::::::::::::
/usr/aset/reports/0723_03:31/usrgrp.rpt
::::::::::::::

*** Begin User And Group Checking ***

Checking /etc/passwd ...

Checking /etc/shadow ...

... end user check.

Checking /etc/group ...

... end group check.

*** End User And Group Checking ***
```

3. Set ASET to run nightly.

```
# /usr/aset/aset -p
======= ASET Execution Log =======

ASET running at security level high

Machine = roo; Current time = 0722_12:35

aset: Using /usr/aset as working directory

ASET execution scheduled through cron.
#
```

Auditing and System Accounting

William Osser paper entitled, "Auditing in the Solaris™ 8 Operating Environment"[11] provides a good discussion of configurating auditing. Auditing was turned on after the software configurator was run. This section implements the recommendations from William Osser paper.

1. To make sure auditing is turned on, start auditing by running bsmconv under level 1. Respond "y" to the prompt "Shall we continue with the conversion now?" For BSM to be enabled, a reboot is done in step 6.

```
# /usr/sbin/init 1
# /etc/security/bsmconv
#
```

2. Pull down the tools developed in William Osser paper available from:

   http://www.sun.com/blueprints/tools/

3. Uncompress, untar, and move the files over into the /etc/security area.

```
# /usr/bin/zcat audit-config.tar.Z | tar xf -
# /usr/bin/cp audit_* /etc/security
#
```

4. Re-enable the Stop-a keyboard sequence. The system Roo is kept in a secure location. The risk of unauthorized keyboard access is minimal. Remove from /etc/system the line "set abort_enable = 0."

5. The system accounting entries exist in the sys cron file. Allow the account sys to run cron jobs and uncomment the lines in sys cron file. This will provide baseline system data every 20 minutes and can be accessed using the sar command.

```
# vi /etc/cron.d/cron.allow
root
sys
# export EDITOR=vi
# /usr/bin/crontab -e sys
0 * * * 0-6 /usr/lib/sa/sa1
20,40 8-17 * * 1-5 /usr/lib/sa/sa1
5 18 * * 1-5 /usr/lib/sa/sa2 -s 8:00 -e 18:01 -i 1200 -A
#
```

6. Reboot the system for changes to take affect.

```
# /usr/sbin/reboot
#
```

Limiting Access with the SunScreen™ Lite Firewall

Sun has made available a free stateful packet screening firewall, SunScreen™ 3.1 Lite. The best source for additional reading on the installation of SunScreen is Sun's document entitled, "SunScreen 3.1 Lite Installation Guide."[12] The document outlines the required packages (page 7) and the software will report on missing packages. See Appendix A-3 for a listing of required Sun OS packages.

This section outlines the installation of SunScreen and the configuration to fit the access requirements of Roo.

SunScreen Command Line Package Installation

Only the command line interface will be used on Roo. Administration of the firewall is to be kept locally and Roo is not going to be running the web interface. Once the firewall is installed, access should be fairly controlled. To keep Roo as secure as possible, Roo will not be running a GUI interface nor will it run any unnecessary services. Sunscreen has proven to be a very good product, but in case of misconfiguration or future exploits being discovered in the software, the standard security practices are going to be maintained, including the use of TCP Wrapper.

The installation procedure by default will attempt to start a Java GUI interface. If Kanga connects to Roo and has the graphics forwarded from Roo to Kanga, then the default installation option is available. What follows is how to install SunScreen in command line mode.

A word of warning, the installation documentation provided by Sun[10] has an additional package listed, "SUNicgSA" (SunScreen online Documentation). This will cause the package numbers to differ.

```
The following packages are available:
 1  NSCPcom     Netscape Communicator
                (sparc) 20.4.70,REV=1999.08.20.17.43
 2  SUNWbdc     SKIP Bulk Data Crypt
                (sparc) 1.5.1
 3  SUNWbdcx    SKIP Bulk Data Crypt (64-bit)
                (sparc) 1.5.1
 4  SUNWdes     SKIP DES Crypto Module
                (sparc) 1.5.1
 5  SUNWdesx    SKIP DES Crypto Module (64-bit)
                (sparc) 1.5.1
 6  SUNWdthj    HotJava Browser for Solaris
                (sparc) 1.1.5,REV=1998.12.03
 7  SUNWdtnsc   Netscape Componentization Support for CDE
                (sparc) 1.0,REV=1999.06.14.15.50
 8  SUNWes      SKIP End System
                (sparc) 1.5.1
 9  SUNWesx     SKIP End System (64-bit)
                (sparc) 1.5.1
10  SUNWfwcnv   SunScreen Firewall conversion
                (sparc) 3.1
11  SUNWhttp    Sun WebServer daemon and supporting binaries
                (sparc) 2.0
12  SUNWicgSA   SunScreen Administration Software
                (sparc) 3.1
13  SUNWicgSF   SunScreen full function
                (sparc) 3.1
14  SUNWicgSM   SunScreen man pages
                (sparc) 3.1
15  SUNWicgSS   SunScreen Firewall
                (sparc) 3.1
16  SUNWkeymg   SKIP Key Manager Tools
                (sparc) 1.5.1
17  SUNWkusup   SKIP U-Support module
                (sparc) 1.5.1
18  SUNWrc2     SKIP RC2 Crypto Module
                (sparc) 1.5.1
19  SUNWrc4     SKIP RC4 Crypto Module
                (sparc) 1.5.1
20  SUNWrc4x    SKIP RC4 Crypto Module (64-bit)
                (sparc) 1.5.1
21  SUNWsman    SKIP Man Pages
                (sparc) 1.5.1
Select package(s) you wish to process (or 'all' to process
all packages). (default: all) [?,??,q]:
```

Because of dependencies, select the following packages: 2-5,8-9,14-19,12, 20

Setting System to Start SunScreen

Run the command "ss_install" to complete the installation. Add "/opt/SUNWicg/SunScreen/bin" to your path. Reboot the system and then run ss-install. Then remove the calls to start the web server interface and remove all the unnecessary files associated with it.

```
# /usr/sbin/init 6
# /opt/SUNWicg/SunScreen/bin/ss_install
# /usr/sbin/init 6
# /usr/bin/vi /etc/rc2.d/S63sunscreen
  remove the line:
        $SS_LIBDIR/run_http start efshttpd
# /usr/bin/vi /opt/SUNWicg/SunScreen/lib/ss_boot
  remove the line:
     $SS_LIBDIR/ssadmserver start >/dev/console 2>&1
# /usr/bin/rm -r /opt/SUNWicg/SunScreen/admin/htdocs
# /usr/bin/rm -r /opt/SUNWicg/SunScreen/admin/cgi-bin
# /usr/bin/rm /usr/bin/ht*
```

Configuring SunScreen

Those familiar with firewall configurations should be able to follow this section. For additional references, Seán Boran has included a section in his documentation, "Hardening Solaris with Yassp."[13]

Setting Hosts Aliases

Refer to Table 2 for a listing of the hosts IPs and what function they will play.

```
# cd /opt/SUNWicg/SunScreen/bin
# ./ssadm edit Initial
edit> add address kanga HOST 172.16.129.138
edit> add address pooh HOST 172.16.129.157
edit> add address eeyore HOST 172.16.129.11
edit> add address tigger HOST 172.16.129.136
edit> save
edit> verify
edit> quit
# ./ssadm activate Initial
#
```

Setting SSH Access

Only SSH access will be allowed to Kanga. During the setup, Kanga pulls down, compiles, and prepares software that Roo needs. Roo needs to be able to secure copy from Kanga.

```
# cd /opt/SUNWicg/SunScreen/bin
# ./ssadm edit Initial
edit> add service ssh SINGLE FORWARD "tcp" PORT 22
edit> add rule ALLOW ssh localhost kanga COMMENT "For secure file
download"
edit> save
edit> verify
edit> quit
# ./ssadm activate Initial
#
```

Setting NTP Access

Throughout the crobbins.org, network machines use NTP to get their clocks in sync with the NT server. Roo needs to have NTP access to the NTP server. Roo also needs to have a file created /etc/inet/ntp.conf. Sun provides a sample ntp.client, which will be used, and

the server set to get its time from Pooh.

```
# cd /opt/SUNWicg/SunScreen/bin
# ./ssadm edit Initial
edit> add rule ALLOW ntp localhost pooh "synchronize localhost clock"
edit> save
edit> verify
edit> quit
# ./ssadm activate Initial
# /usr/sbin/cp  /etc/inet/ntp.client  /etc/inet/ntp.conf
# /usr/bin/vi /etc/inet/ntp.conf
       server 172.16.129.157
# /etc/rc2.d/S74xntpd start
#
```

Setting Syslog Server Access

Access needs to be granted to the clients, which will be sending Roo their log files. In addition, allow Roo to check if the clients are up by being able to use ping. Check that the startup file (/etc/rc.2d/S74syslog) has the command "/usr/sbin/syslogd –t".

```
# cd /opt/SUNWicg/SunScreen/bin
# ./ssadm edit Initial
edit> add address syslog-clients GROUP kanga eeyore tigger
edit> add rule ALLOW syslog syslog-clients localhost COMMENT "Syslog
operations"
edit> add rule ALLOW pint localhost syslog-clients COMMENT "BB check clients
are up"
edit> save
edit> verify
edit> quit
# ./ssadm activate Initial
#
```

Setting DNS Access

Since Roo's function is to collect and analyze log files, being able to query the DNS server to resolve IP addresses is important. Roo needs access to the DNS server.

```
# cd /opt/SUNWicg/SunScreen/bin
# ./ssadm edit Initial
edit> add rule ALLOW dns localhost pooh COMMENT "Grant localhost access to
DNS"
edit> save
edit> verify
edit> quit
# ./ssadm activate Initial
# /usr/bin/vi /etc/resolv.conf
     domain crobbins.org
     nameserver 172.16.129.157
#
```

Setting Logging Access

It is important to monitor who's attempting to access your machine. In order to do this, Sunscreen must be told to log.

```
# cd /opt/SUNWicg/SunScreen/bin
# ./ssadm edit Initial
edit> add interface "hme0" SCREEN "roo" ROUTING "roo_hme0" LOG SUMMARY ICMP
NONE
edit> save
edit> verify
edit> quit
# ./ssadm activate Initial
#
```

Once telling SunScreen to log, make sure to review the logs.

```
# cd /opt/SUNWicg/SunScreen/bin
# ./ssadm log get | ./ssadm logdump -i -
12879     hme0 (256: deny rule or no pass rule)195022.19353
          scan1.crobbins.org -> 172.16.129.159 UDP  D=138 S=138
LEN=196
12880     hme0 (256: deny rule or no pass rule)195034.14694
          kares.crobbins.org -> 172.16.129.255 UDP  D=137 S=137
LEN=58
12881     hme0 (256: deny rule or no pass rule)195034.14743
          genesis -> 172.16.129.255 UDP D=137 S=137 LEN=58
12882     hme0 (256: deny rule or no pass rule)195047.14351
          spock -> 172.16.129.255 UDP D=138 S=138 LEN=227
12883     hme0 (256: deny rule or no pass rule)195047.14392
          kares.crobbins.org -> 172.16.129.255 UDP D=138 S=138
LEN=227
12884     hme0 (256: deny rule or no pass rule)195047.14399
          genesis -> 172.16.129.255 UDP D=138 S=138 LEN=227
#
```

Logging Functionality

The machines from the CRobinns.org will send their system log files to Roo. Roo will be using the Psionic Logcheck software to scan for occurrences within the log files. This section will cover the configuration and installation of software involved in performing log analysis.

Syslog Server

Setting a machine to act as a syslog server requires modifications to be made on both the server and client side. The server must know to accept logs from other machines while the clients must know where to send their logs.

On the server

Sun has provided a sample syslog.conf.server file. Copy the syslog.conf.server over the syslog.conf file. The logs are going to be broken up into separate files, which will require modification of the syslog.conf file. Separate log files allow greater flexibility when dealing with the program Psionic Logcheck. After making the modifications, restart the syslog server.

```
# /usr/bin/cp /etc/syslog.conf.server /etc/syslog.conf
# /usr/bin/vi /etc/syslog.conf
replace:

   *.err;kern.notice;auth.info                    @loghost
   *.info;mail.none                               /var/adm/messages
   mail.debug                       /var/log/syslog
   user.err                         /dev/sysmsg
   user.err                         /var/adm/messages
   user.alert                       `root, operator'
   user.emerg                       *

with:

   kern.info               ifdef(`LOGHOST', /var/log/kernlog, @loghost)
   user.info               ifdef(`LOGHOST', /var/log/userlog, @loghost)
   mail.info               ifdef(`LOGHOST', /var/log/maillog, @loghost)
   daemon.info             ifdef(`LOGHOST', /var/log/daemonlog,
@loghost)
   auth.info               ifdef(`LOGHOST', /var/log/authlog, @loghost)
   lpr.info                ifdef(`LOGHOST', /var/log/lprlog, @loghost)
   news,uucp.info          ifdef(`LOGHOST', /var/log/newslog, @loghost)
   cron.info               ifdef(`LOGHOST', /var/log/cronlog, @loghost)
   local0,local1.info      ifdef(`LOGHOST', /var/log/local0log,
@loghost)
   local2,local3,local4.info  ifdef(`LOGHOST', /var/log/local2log,
@loghost)
   local5,local6,local7.info  ifdef(`LOGHOST', /var/log/local5log,
@loghost)
   *.err                   ifdef(`LOGHOST', /var/log/alertlog, @loghost)
# /etc/rc2.d/S74syslog stop
# /etc/rc2.d/S74syslog start
#
```

On the Client

On the client side, adding an entry to the /etc/hosts file with an alias "loghost" and restarting the syslog service will start the client logging to Roo.

```
# /usr/bin/vi /etc/hosts
  kanga loghost
# /etc/rc2.d/S74syslog stop
# /etc/rc2.d/S74syslog start
#
```

Psionic Logcheck

Logcheck will examine system log files for unusual activities. The user can define what log files are checked and how they are checked.

1. Obtain the Psionic Logcheck software, Craig Rowland's PGP key, and Logcheck's signature from:

   http://www.psionic.com/download

2. Import Craig Rowland's public key and check the software against the signature.

```
# cd /local/software
# /usr/local/bin/gpg --import crowland.asc
gpg: /local/home/jgerber/.gnupg/secring.gpg: keyring created
gpg: /local/home/jgerber/.gnupg/pubring.gpg: keyring created
gpg: key 98ABFE7D: public key imported
gpg: /local/home/jgerber/.gnupg/trustdb.gpg: trustdb created
gpg: Total number processed: 1
gpg:               imported: 1  (RSA: 1)
# /usr/local/bin/gpg --verify logcheck-1.1.1.tar.gz.asc logcheck-
1.1.1.tar.gz
gpg: Signature made Wed Dec 01 22:18:22 1999 EST using RSA key ID 98ABFE7D
gpg: Good signature from "Craig H. Rowland <crowland@psionic.com>"
Could not find a valid trust path to the key.  Let's see whether we
can assign some missing owner trust values.

No path leading to one of our keys found.

gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the
owner.
gpg: Fingerprint: AD 1E A8 42 8D AA DA 18  D1 04 B4 B1 23 13 6F 9C
#
```

3. Psionic Logcheck requires GNU grep. Pull the package from:

   http://www.sunfreeware.com/programlistsparc8.html#grep

4. Add the GNU grep package.

```
# cd /local/software
# /usr/sbin/pkgadd -d grep-2.4.2-sol8-sparc-local

The following packages are available:
  1  SMCgdbm     gdbm
                 (sparc) 1.8.0

Select package(s) you wish to process (or 'all' to process
all packages). (default: all) [?,??,q]: 1
#
```

5. Configure and compile the software. Modify the file logcheck.sh so it contains "SYSADMIN=sysadmin-page." The /usr directory, which is where Logcheck writes its temporary files by default, is going to be mounted read-only on Roo. Modify logcheck.sh so "TMPDIR=/local/etc/tmp." Add to logcheck.sh the system log file names. Edit the file Makefile so it contains "CC=gcc".

```
# cd /local/software
# /usr/bin/gunzip -c logcheck-1.1.1.tar.gz | /usr/bin/tar xf -
# cd logcheck-1.1.1
# /usr/bin/mkdir -p -m 700 /local/etc/tmp
# /usr/bin/vi ./systems/sun/logcheck.sh
SYSADMIN=sysadmin-page
TMPDIR=/local/etc/tmp
# SunOS, Sun Solaris 2.5
$LOGTAIL /var/log/kerlog > $TMPDIR/check.$$
$LOGTAIL /var/log/authlog >> $TMPDIR/check.$$
$LOGTAIL /var/log/alertlog >> $TMPDIR/check.$$
$LOGTAIL /var/log/userlog >> $TMPDIR/check.$$
$LOGTAIL /var/log/maillog >> $TMPDIR/check.$$
$LOGTAIL /var/log/daemonlog >> $TMPDIR/check.$$
$LOGTAIL /var/log/authlog >> $TMPDIR/check.$$
$LOGTAIL /var/log/lprlog >> $TMPDIR/check.$$
$LOGTAIL /var/log/newslog >> $TMPDIR/check.$$
$LOGTAIL /var/log/cronlog >> $TMPDIR/check.$$
$LOGTAIL /var/log/local0log >> $TMPDIR/check.$$
$LOGTAIL /var/log/local2log >> $TMPDIR/check.$$
$LOGTAIL /var/log/local5log >> $TMPDIR/check.$$
# /usr/bin/vi Makefile
CC=gcc
# /usr/local/bin/make sun
# /usr/bin/crontab -e
0,5,10,15,20,25,30,35,40,45,50,55 * * * * /usr/local/etc/logcheck.sh
#
```

6. Check that Logcheck runs.

```
# /usr/local/etc/logcheck.sh
#
```

7. Examine the results of Logcheck. A sample message could look like:

```
Security Violations
=-------=-------=
Jun 25 17:00:00 tigger.crobbins.org su: 'su monitor' succeeded for root on
/dev/???
Jun 25 17:08:47 roo sshd[3619]: [ID 800047 auth.info] Failed password for
jgerber
   from 172.16.129.138 port 1022
Jun 25 17:08:55 roo sshd[3642]: [ID 800047 auth.info] Failed password for
jgerber
   from 172.16.129.138 port 1020

Unusual System Events
=-------=-------=-------=
Jun 25 17:00:00 tigger.crobbins.org su: 'su monitor' succeeded for root on
    /dev/???
Jun 25 17:00:03 tigger.crobbins.org last message repeated 1 time
Jun 25 17:01:53 tigger.crobbins.org ID[RICHPse.monlog.2000]: Network state
green
   entered,    Action: No worries, mate
Jun 25 17:03:53 tigger.crobbins.org ID[RICHPse.monlog.2000]: Network state red
   entered, Action: Add more or faster nets
```

Proper configuration will depend on the traffic received at the site. Modify the files logcheck.hacking, logcheck.violations, and logcheck.violations.ignore under /usr/local/etc based on events observed at the site. The unusual events section might be considered less critical then the security violations section. Modifications can be made to the /usr/local/etc/logcheck.sh file, changing how logcheck.sh handles the different sections of the report. The system administrators could be paged on security violations while unusual events are sent as e-mail.

Other possible modifications are to have multiple versions of Logcheck run on different log files, allowing different schedules to be set up for different files. For example, the kernel, authorization, and alert log might be checked every five minutes while the mail log is checked less frequently. The Psionic license does grant permission to modify the source for personal use only. Not being sure if inclusion of the modifications in this paper would be considered public use, modifications are left as an exercise to the reader.

Rotating Logs

Jean Chouanard has developed a software package, PARCdaily, which will do log rotations, backups of system files, and checks on partition sizes and packages installed. PARCdaily will require GNU Revision Control System (RCS).

1. Pull down Jean Chouanard's public key, the software package PARCdaily, and the software's signature file:

   http://www.yassp.org/download.html

2. Import Jean Chouanard's public key and check the PARCdaily software against its signature.

```
# cd /local/software
# /usr/local/bin/gpg --import chouanard.asc
gpg: key 160B187B: public key imported
gpg: Total number processed: 1
gpg:               imported: 1  (RSA: 1)
# /usr/local/bin/gpg --verify parcdaily.Z.asc parcdaily.Z
gpg: Signature made Mon Nov 20 00:55:01 2000 EST using RSA key ID 160B187B
gpg: Good signature from "Jean Chouanard <jean@h2tp.com>"
gpg:                 aka "Jean Chouanard <chouanard@parc.xerox.com>"
gpg:                 aka "Jean Chouanard <jean@cinops.xerox.com>"
gpg:                 aka "Jean Chouanard <jean@parc.xerox.com>"
Could not find a valid trust path to the key.  Let's see whether we
can assign some missing owner trust values.

No path leading to one of our keys found.

gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the
owner.
gpg: Fingerprint: 7A 3D CC 65 B6 9A 80 90  95 BD B4 BB 75 B6 00 DB

#
```

3. The PARCdaily package requires RCS package to be installed. Pulled down RCS from:

   http://www.sunfreeware.com/programlistsparc8.html#rcs

4. Install the RCS package.

```
# /usr/sbin/pkgadd -d rcs-5.7-sol8-sparc-local

The following packages are available:
  1  SMCrcs     rcs
                (sparc) 5.7

Select package(s) you wish to process (or 'all' to process
all packages). (default: all) [?,??,q]: 1
#
```

5. Uncompress and install the PARCdaily package.

```
# cd /local/software
# /usr/bin/uncompress parcdaily.Z
# /usr/sbin/pkgadd -d parcdaily

The following packages are available:
  1  PARCdaily     DailyCronJob
                   (sparc,i386) 1.12 $Id: pkginfo,v 1.16 2000/11/20 00:12:53 chouanar
Exp $

Select package(s) you wish to process (or 'all' to process
all packages). (default: all) [?,??,q]: 1
# /usr/bin/uncompress gnurcs_sparc.Z
# /usr/sbin/pkgadd -d gnurcs_sparc

The following packages are available:
  1  GNUrcs     GNU rcs and diffutils
                (sparc) 5.7

Select package(s) you wish to process (or 'all' to process
all packages). (default: all) [?,??,q]: 1
#
```

6. When PARCdaily is installed, it will set the program /opt/local/sbin/daily to run each night at 23:58. Confirm that the changes were made.

```
# crontab -e
0,15,30,45 * * * * /usr/local/etc/logcheck.sh
0 * * * * /usr/lib/sendmail -q
0 0 * * * /usr/aset/aset  -d /usr/aset
0 * * * * /usr/sbin/audit -n
# Daily logs rotation / files archiving / files checks #PARCdaily
58 23 * * * /opt/local/sbin/daily  #PARCdaily
#
```

7. Modify /opt/local/sbin/daily to use "/usr/bin/gzip" instead of "/opt/local/bin/gzip."

```
# /usr/bin/vi /opt/local/sbin/daily
GZIP=/usr/bin/gzip
#
```

Communications with Administrators

In order to allow rapid notification, the syslog box has been configured with a modem. In order to use the modem, the system needs to be able to physically communicate with the modem and have software that allows alphanumeric messages to be sent.

Terminal Setup

A very good source for terminal setup information for Solaris is Celeste Stokely, "Celeste's Tutorial on Solaris 2.x Modems & Terminals."[14] Make sure ownership and group permissions are set. The device /dev/cua/a is where the modem is attached and QuickPage will be running as user uucp. Set the incoming device, /dev/term/a so incoming calls will be ignored.

```
# chown uucp /dev/cua/a; chgrp tty /dev/cua/a
# chown root /dev/term/a; chgrp tty /dev/term/a
#
```

QuickPage

1. Pull down the latest QuickPage software from:

   http://www.qpage.org/download.html

2. Uncompress and untar the QuickPage software.

```
# cd /local/software
# /usr/bin/uncompress qpage-3.3.tar.Z
# /usr/bin/tar xf qpage-3.3.tar
#
```

3. Modify the file config.input so DAEMON_USER is set to "uucp". Run configure, make, and install.

```
# cd /local/software/qpage-3.3
# /usr/bin/vi config.input
# ./configure
# /usr/local/bin/make
# /usr/local/bin/make install
# /usr/bin/chmod uog+rx  /usr/local/bin/qpage
#
```

4. Modify the configuration file adding pager number for the administrators. Create startup file. Add QuickPage to the TCP Wrapper access control.

```
# /usr/bin/vi /etc/qpage.cf
        administrator=jgerber@kanga
        queuedir=/var/spool/qpage

        identtimeout=5
        snpptimeout=60

        modem=ttya device=/dev/cua/a
        modem=ttyb device=/dev/cua/b
        service=default
                device=ttyb
                baudrate=1200
                parity=even
                allowpid=yes
                maxtries=6
                phone=3019999
                maxmsgsize=256
        pager=jgerber
                text=John_J_Gerber
                pagerid=3016577
                pagerid=3016577
                service=default

        pager=deppd
                text=Dennis_Depp
             pagerid=3016652
             service=default

        group=sysadmin
                text=System_Administrator
                member=jgerber/Any0000-2359
                member=deppd/Any0800-1700
# /usr/bin/vi /etc/qpage.servers
  localhost
# /usr/bin/cp S99qpage /etc/init.d/qpage
# /usr/bin/chmod u+x  /etc/init.d/qpage
# /usr/bin/ln -s /etc/init.d/qpage /etc/rc3.d/S99qpage
# /usr/bin/vi /etc/hosts.allow
   qpage: localhost,172.16.129.178,127.0.0.1
#
```

5. Add entries to the /etc/aliases file (only line being added are shown).

```
# /usr/bin/vi /etc/aliases
sysadmin-page: "|/usr/local/bin/qpage -l 0 -m -p sysadmin"
jgerber-page: "|/usr/local/bin/qpage -l 0 -m -p jgerber"
deppd-page: "|/usr/local/bin/qpage -l 0 -m -p deppd"
# /usr/sbin/newaliases
#
```

6. Start QuickPage.

```
# /etc/rc3.d/S99qpage start
#
```

Securing Logs

The log files that are not being actively written to and analyzed can be placed in an archive directory. The archive area should regularly be encrypted to add another layer of security. This section demonstrates how to use the GNU Privacy Guard software to secure log files. Keeping the log files in an archive area allows the logs to exist for a time during which the system administrator can access them. The best security is to get the log files burned onto a CD and place the CD in a secure location. Using the Solaris commands cdrw and mkisofs to write data out to a CD is demonstrated.

Encrypting Logs with GNU Privacy Guard

Encrypting of the log files can be done using GNU Privacy Guard. Currently, the unencrypted log files are left in place for the convenience of the system administrator. Dump out the SunScreen packet logs to a file in the /var/oldlogs directory. The files modified in the past day will be tarred together and encrypted. The process can be automated with a shell script which would be set to run after the /opt/local/bin/daily script. For this script to work, the recipient key will need to be in root's keyring.

```
# /usr/bin/mkdir -p -m 700 /local/admin/bin
# /usr/bin/mkdir -p -m 700 /local/secarch
# /usr/bin/vi /local/admin/bin/seclogs.sh
#!/sbin/sh
LD_LIBRARY_PATH=/usr/local/lib
PATH=/usr/bin:/usr/sbin:/opt/local/bin
export LD_LIBRARY_PATH
export PATH
/opt/SUNWicg/SunScreen/bin/ssadm log get > /var/oldlogs/ssadm_`date
'+%y%m%d'`
/opt/SUNWicg/SunScreen/bin/ssadm log clear
/usr/bin/gzip /var/oldlogs/ssadm_`date '+%y%m%d'`
/usr/bin/find /var/oldlogs /var/audit /usr/aset/reports -mtime -1 -type f |
\
   /usr/local/bin/tar -cf /local/secarch/`date '+%y%m%d'`.log.tar -T -
/usr/local/bin/gpg --output /local/secarch/`date '+%y%m%d'`.gpg --encrypt \
  --recipient "John Gerber <John.J.Gerber@crobbins.org>"  \
  /local/secarch/`date '+%y%m%d'`.log.tar
/usr/bin/rm /local/secarch/`date '+%y%m%d'`.log.tar
# /usr/bin/chmod u+x /local/admin/bin/seclogs.sh
# /usr/bin/crontab -e
0,15,30,45 * * * * /usr/local/etc/logcheck.sh
0 * * * * /usr/lib/sendmail -q
0 0 * * * /usr/aset/aset  -d /usr/aset
0 * * * * /usr/sbin/audit -n
# Daily logs rotation / files archiving / files checks #PARCdaily
58 23 * * * /opt/local/sbin/daily ; /local/admin/bin/seclogs.sh
#
```

GNU tar provides greater functionality, which is used in seclogs.sh. Pull down the GNU tar package from:

http://www.sunfreeware.com/programlistsparc8.html#tar

Add the GNU tar package.

.

```
# /usr/sbin/pkgadd -d tar-1.13-sol8-sparc-local

The following packages are available:
  1  SMCtar     tar
                (sparc) 1.13

Select package(s) you wish to process (or 'all' to process
all packages). (default: all) [?,??,q]: 1
#
```

Writing Logs to CD

The following command will write any files existing in the secure archive area, /local/secarch out to the CD-RW drive. In this example, the volume name "SL_July_2001" is used.

```
# mkisofs -J -r -L -V SL_July_2001 -o /vcd/july2001.iso
/local/secarch
# cdrw -d /dev/rdsk/c1t4d0s2 -i  /vcd/july2001.iso
#
```

Mirroring the System

Mirroring does protect against drive failure, but it can also be used as a security tool. Solstice DiskSuite (SDS), which is freely available from Sun, will mirror drives. The method described in this section avoids the additional packages of SDS. It has the added benefit of protection against unwanted modifications. If a malicious cracker manages to get the "rm –r /" command to run on a SDS system, SDS will just mirror the modifications as they are made until the system crashes. The method described below can avoid this problem. Mirroring can be done as frequently as desired.

Additional steps could be taken to try to hide the existence of backup partitions from prying eyes and burying the execution of the program within other innocent looking cron entries. Take the security through obscurity idea to any level desired. This section introduces the idea of mirroring of the system onto a secondary drive as another layer of protection. Roo is a relatively small system, so complete mirroring is possible.

This section follows closely the Seán Boran article entitled, "A Tool for Cold Mirroring of Solaris System Disks."[15]

1. Obtain the mirror.boot.sh script from:

   http://www.securityportal.com/research/solaris/mirror_boot.sh

2. Setup secondary disk so it is identical to the primary and create secondary drives file systems.

```
# /usr/sbin/prtvtoc /dev/rdsk/c0t0d0s2 | /usr/sbin/fmthard -n mirror -s -
\
  /dev/rdsk/c0t1d0s2
fmthard:  New volume table of contents now in place.

# for i in 0 1 3 4 5 6 7
> do
> /usr/sbin/newfs /dev/rdsk/c0t1d0s${i}
> done
#
```

3. Modify /etc/vfstab, adding the new file systems under root directory "newroot." Mount the newroot file system and make mount points for the other newroot partitions. Mount and immediately umount the mirrored partitions to verify that the partitions are mountable.

```
# /usr/bin/vi /etc/vfstab
#device         device          mount           FS      fsck    mount   mount
#to mount       to fsck         point           type    pass    at boot options
#
#/dev/dsk/c1d0s2 /dev/rdsk/c1d0s2 /usr          ufs     1       yes     -
fd      -       /dev/fd fd      -       no      -
/proc   -       /proc   proc    -       no      -
/dev/dsk/c0t0d0s3       -       -       swap    -       no      logging
/dev/dsk/c0t0d0s0  /dev/rdsk/c0t0d0s0      /       ufs     1    no  -
/dev/dsk/c0t0d0s4  /dev/rdsk/c0t0d0s4      /usr    ufs     1    no  ro
/dev/dsk/c0t0d0s1  /dev/rdsk/c0t0d0s1      /var    ufs     1    no
logging,nosuid,noatime
/dev/dsk/c0t0d0s6  /dev/rdsk/c0t0d0s6      /local  ufs     2    yes logging,nosuid
/dev/dsk/c0t0d0s7  /dev/rdsk/c0t0d0s7      /vcd    ufs     2    yes -
/dev/dsk/c0t0d0s5  /dev/rdsk/c0t0d0s5      /opt    ufs     2    yes logging
swap    -       /tmp    tmpfs   -       yes     size=100m
/dev/dsk/c0t1d0s0  /dev/rdsk/c0t1d0s0      /newroot        ufs  1  no  logging
/dev/dsk/c0t1d0s4  /dev/rdsk/c0t1d0s4      /newroot/usr    ufs  1  no  logging
/dev/dsk/c0t1d0s1  /dev/rdsk/c0t1d0s1      /newroot/var    ufs  1  no
logging,nosuid,noatime
/dev/dsk/c0t1d0s6  /dev/rdsk/c0t1d0s6      /newroot/local  ufs  2  no  logging
/dev/dsk/c0t1d0s5  /dev/rdsk/c0t1d0s5      /newroot/opt    ufs  2  no  logging
# /usr/sbin/mount  /newroot
# for i in usr var local opt
> do
> /usr/bin/mkdir -m 777 /newroot/${i}
> /usr/sbin/mount /newroot/${i}
> /usr/sbin/umount /newroot/${i}
> done
#
```

4. Create the file /etc/vfstab.newroot, which will be copied into the mirrored root partition as /etc/vfstab.

```
# /usr/bin/cp /etc/vfstab /etc/vfstab.newroot
# /usr/bin/vi /etc/vfstab.newroot
#device         device          mount           FS      fsck    mount   mount
#to mount       to fsck         point           type    pass    at boot options
#
#/dev/dsk/c1d0s2 /dev/rdsk/c1d0s2 /usr          ufs     1       yes     -
fd      -       /dev/fd fd      -       no      -
/proc   -       /proc   proc    -       no      -
/dev/dsk/c0t1d0s3   -       -       swap    -       no      -
/dev/dsk/c0t1d0s0   /dev/rdsk/c0t0d0s0   /       ufs   1   no   logging
/dev/dsk/c0t1d0s4   /dev/rdsk/c0t0d0s4   /usr    ufs   1   no   ro
/dev/dsk/c0t1d0s1   /dev/rdsk/c0t0d0s1   /var    ufs   1   no
logging,nosuid,noatime
/dev/dsk/c0t1d0s6   /dev/rdsk/c0t0d0s6   /local  ufs   2   yes
logging,nosuid,noatime
/dev/dsk/c0t1d0s5   /dev/rdsk/c0t0d0s5   /opt    ufs   2   yes  ro
swap    -       /tmp    tmpfs   -       yes     -
```

5. Modify /secure/mirror_boot.sh as required. The modified mirror.boot.sh script (renamed /secure/mirror_boot.sh) appear in the A-4 Appendix. Change the variable "DEBUG" to "1" and validate the configuration. When in debug mode, the program will not copy data. After confirming the configuration, change the "DEBUG" variable back to "0" and run the program.

```
# vi /secure/mirror_boot.sh
# /secure/mirror_boot.sh
DEBUG mode on
mkdir -p /newroot/proc
mkdir -p /newroot/var
mkdir -p /newroot/var/run
mkdir -p /newroot/opt
mkdir -p /newroot/usr
mkdir -p /newroot/local
mkdir -p /newroot/oldroot
----- Backup /var /opt /usr /local to /newroot----
Logfile: /var/tmp/f1937 on Wed Jul 18 12:00:36 EDT 2001

Backing up root filesystem at Wed Jul 18 12:00:36 EDT 2001 to /newroot
find . -xdev -print | cpio -pdmu /newroot at Wed Jul 18 12:00:36 EDT 2001
DEBUG MODE: no data actually copied
make mount points /proc  /var /var/run /opt /usr /local /oldroot ..
mounting /newroot/var
Backing up /var at Wed Jul 18 12:00:36 EDT 2001
find . -xdev -print | cpio -pdmu /newroot/var at Wed Jul 18 12:00:36 EDT
2001
DEBUG MODE: no data actually copied
mounting /newroot/opt
Backing up /opt at Wed Jul 18 12:00:36 EDT 2001
find . -xdev -print | cpio -pdmu /newroot/opt at Wed Jul 18 12:00:36 EDT
2001
DEBUG MODE: no data actually copied
mounting /newroot/usr
Backing up /usr at Wed Jul 18 12:00:36 EDT 2001
find . -xdev -print | cpio -pdmu /newroot/usr at Wed Jul 18 12:00:36 EDT
2001
DEBUG MODE: no data actually copied
mounting /newroot/local
Backing up /local at Wed Jul 18 12:00:36 EDT 2001
find . -xdev -print | cpio -pdmu /newroot/local at Wed Jul 18 12:00:36 EDT
2001
DEBUG MODE: no data actually copied

Filesystem            kbytes    used   avail capacity  Mounted on
/dev/dsk/c0t0d0s0     246463   46915  174902    22%    /
/dev/dsk/c0t0d0s4    1015542  168926  785684    18%    /usr
/dev/dsk/c0t0d0s1    2052750   81839 1909329     5%    /var
/dev/dsk/c0t0d0s6   11091916 1612095 9368902    15%    /local
/dev/dsk/c0t0d0s7    2052750  636423 1395800    32%    /vcd
/dev/dsk/c0t0d0s5    1015542  107690  846920    12%    /opt
/dev/dsk/c0t1d0s0     246463    1404  220413     1%    /newroot
/dev/dsk/c0t1d0s1    2052750   83455 1907713     5%    /newroot/var
/dev/dsk/c0t1d0s5    1015542  108682  845928    12%    /newroot/opt
/dev/dsk/c0t1d0s4    1015542  168806  785804    18%    /newroot/usr
/dev/dsk/c0t1d0s6   11091916 1622541 9358456    15%    /newroot/local

Prom boot order is: boot-device=disk0

Copy over /etc/vfstab.newroot....
umount /newroot/{/var /opt /usr /local}
Install boot block on /dev/rdsk/c0t1d0s0

Finished at Wed Jul 18 12:00:37 EDT 2001

This email was generated by roo:/secure/mirror_boot.sh

# vi /secure/mirror_boot.sh
# /secure/mirror_boot.sh
#
```

6. The EEPROM has a definition of disk1. Set the boot-device so the machine will boot off the mirror in the case of the primary disk (disk0) failing.

```
# eeprom boot-device "disk0 disk1"
#
```

7. Configure the system to run the mirror software nightly during the weekdays.

```
# crontab -e
0,15,30,45 * * * * /usr/local/etc/logcheck.sh
0 * * * * /usr/lib/sendmail -q
0 0 * * * /usr/aset/aset  -d /usr/aset
0 * * * * /usr/sbin/audit -n
58 23 * * * /opt/local/sbin/daily ; /local/admin/bin/seclogs.sh
0 23 * * 1-5 /secure/mirror_boot.sh
#
```

8. Bring the system to the "ok" prompt and try booting off the mirror drive.

```
ok boot disk1
ok
```

Checking Security Score

The Center for Internet Security has developed a tool for producing security scores and benchmarks. This provides a way for sites to evaluate and improve the security of their systems. In this section, the Solaris Benchmark & Scoring Tools will be used to provide a measure of Roo's security.

1. Pull down the Solaris Benchmark & Scoring Tools V1.01, the PGP key used to sign the tool, and the PGP signature of the tool from:

   http://www.cisecurity.org/bench_solaris.html

2. Load the key and check the PGP signature and MD5 checksum.

```
# cd /local/software
# /usr/local/bin/gpg --import cis.asc
gpg: key B3E3832C: public key imported
gpg: Total number processed: 1
gpg:               imported: 1
# /usr/local/bin/gpg --verify cis.tar.Z.asc cis.tar.Z
gpg: Signature made Wed Jul 18 02:02:29 2001 EDT using DSA key ID B3E3832C
gpg: Good signature from "CIS Solaris Benchmark v1.0 <sol-
bench@cisecurity.org>"
Could not find a valid trust path to the key.  Let's see whether we
can assign some missing owner trust values.

No path leading to one of our keys found.

gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the
owner.
gpg: Fingerprint: 4AE8 7A53 FA93 0E80 AF95  6583 5DED 0444 B3E3 832C
# /usr/local/bin/md5 cis.tar.Z
MD5 (cis.tar.Z) = 093533b6bccecb82e5c20b6e4384d019
#
```

3. Uncompress, untar, and install the CISscan package.

```
# /usr/bin/uncompress cis.tar.Z
# /usr/bin/tar xf cis.tar
# cd cis
# /usr/sbin/pkgadd -d CISscan

The following packages are available:
  1  CISscan     CIS Host-based Scanning Tool
                 (sparc) 1.0.1

Select package(s) you wish to process (or 'all' to process
all packages). (default: all) [?,??,q]: 1
#
```

4. Run the cis-scan software.

```
# /opt/CIS/cis-scan
*** CIS Ruler v1.0.1 ***
Copright 2001, The Center for Internet Security

Placing logs in /opt/CIS/cis-ruler-log.20010723-05:10:31

Investigating system...this will take a few minutes...
...Found Solaris version 5.8...
...Reading and caching /etc/passwd and /etc/shadow...
...Reading and caching /etc/shells...
...Reading and parsing /etc/vfstab for later use...
...Reading and caching /etc/system...
...Parsing inetd.conf...
...Cataloging rc scripts...
...Checking listening TCP ports via netstat...
...Checking listening UDP ports via netstat...

Beginning system evaluation...
Now a final check for Set-UID and Set-GID programs-- this can take a
whole
lot of time if you have a large filesystem.  Your score if there are
no extra SUID/SGID programs found will be 9.67 / 10.00 .  If there are
extra SUID/SGID programs, your score will be 9.51 / 10.00 .

   You can hit CTRL-C at any time to stop at this remaining step.

find: cannot open /newroot/usr: No such file or directory
find: cannot open /newroot/usr: No such file or directory
find: cannot open /newroot/var: No such file or directory
find: cannot open /newroot/var: No such file or directory
find: cannot open /newroot/local: No such file or directory
find: cannot open /newroot/local: No such file or directory
find: cannot open /newroot/opt: No such file or directory
find: cannot open /newroot/opt: No such file or directory

        Rating = 9.67 / 10.00

----------------------------------------
To learn more about the results, do the following:

   All results/diagnostics:
        more /opt/CIS/cis-ruler-log.20010723-05:10:31
   Positive Results Only:
        egrep "^Positive" /opt/CIS/cis-ruler-log.20010723-05:10:31
   Negative Results Only:
        egrep "^Negative" /opt/CIS/cis-ruler-log.20010723-05:10:31

For each item that you score or fail to score on, please reference the
corresponding item in the CIS Benchmark Document.


   For additional instructions/support, please reference the CIS web
page:

                        http://www.cisecurity.org


# /usr/bin/egrep "^Negative" /opt/CIS/cis-ruler-log.20010723-05:10:31
Negative: 7.4 Non-root accounts are in cron.allow.
Negative: 7.4 Non-root accounts are in cron.allow.
#
```

5. Check for any negatives results.

```
# /usr/bin/egrep "^Negative" /opt/CIS/cis-ruler-log.20010723-05:10:31
Negative: 7.4 Non-root accounts are in cron.allow.
#
```

The non-root account in the cron.allow files is the sys account used for doing the system accounting.

Building Future Boxes

Creating Bootable JumpStart Installation CD-ROM

An installation on a secure network segment allows for the use of JumpStart. John Howard from Sun has written several articles that are available from the Sun blueprints area (https://www.sun.com/blueprints). Alex Noordergraaf has also written a general and concise introduction article for Sun entitled "Building a Jumpstart Infrastructure."[16] Sun's "Solaris 8 Advance Installation Guide"[6] can serve as a reference. If the network is not secure, JumpStart can still be used by building a bootable Jumpstart installation CD-ROM. This method is described in detail by John Howard's article entitled, "Building a Bootable JumpStart™ Installation CD-ROM."[17] This method helps reduce the time and possibilities of errors when installing Solaris on additional machines.

1. Place the "Solaris 8 Software 1 of 2" CD into the CD driver.
2. Ensure the Volume Manger is not running with the command:

```
# /etc/init.d/volmgt stop
#
```

3. Examine the Volume Table of Contents (VTOC):

```
# /usr/sbin/prtvtoc /dev/dsk/c0t2d0s2
* /dev/dsk/c0t2d0s0 partition map
*
* Dimensions:
*     512 bytes/sector
*     640 sectors/track
*       1 tracks/cylinder
*     640 sectors/cylinder
*    2048 cylinders
*    2048 accessible cylinders
*
* Flags:
*   1: unmountable
*  10: read-only
*
* Unallocated space:
*       First     Sector    Last
*       Sector     Count    Sector
*     1292800      2560   1295359
*     1303040      7680   1310719
*
*                          First     Sector    Last
* Partition  Tag  Flags    Sector     Count    Sector  Mount
Directory
       0      4    10          0   1120000   1119999
       1      2    10    1120000    172800   1292799
       2      0    00    1292800      2560   1295359
       3      0    00    1295360      2560   1297919
       4      0    00    1297920      2560   1300479
       5      0    00    1300480      2560   1303039
#
```

4. Examine the drives:

```
# /usr/sbin/format
Searching for disks...done


AVAILABLE DISK SELECTIONS:
       0. c0t0d0 <ST320420A cyl 39533 alt 2 hd 16 sec 63>
          /pci@1f,0/pci@1,1/ide@3/dad@0,0
       1. c0t1d0 <ST320420A cyl 39533 alt 2 hd 16 sec 63>
          /pci@1f,0/pci@1,1/ide@3/dad@0,0
#
```

5. Create the partition that will be used as the virtual CD:

```
# /usr/sbin/newfs -m 1 /dev/rdsk/c0t0d0s7
newfs: /dev/rdsk/c0t0d0s7 last mounted as /vcd
newfs: construct a new file system /dev/rdsk/c0t0d0s7: (y/n)? y
/dev/rdsk/c0t0d0s7:     4195296 sectors in 4162 cylinders of 16 tracks, 63
sectors
        2048.5MB in 66 cyl groups (64 c/g, 31.50MB/g, 5312 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
 32, 64608, 129184, 193760, 258336, 322912, 387488, 452064, 516640, 581216,
 645792, 710368, 774944, 839520, 904096, 968672, 1032224, 1096800, 1161376,
 1225952, 1290528, 1355104, 1419680, 1484256, 1548832, 1613408, 1677984,
 1742560, 1807136, 1871712, 1936288, 2000864, 2064416, 2128992, 2193568,
 2258144, 2322720, 2387296, 2451872, 2516448, 2581024, 2645600, 2710176,
 2774752, 2839328, 2903904, 2968480, 3033056, 3096608, 3161184, 3225760,
 3290336, 3354912, 3419488, 3484064, 3548640, 3613216, 3677792, 3742368,
 3806944, 3871520, 3936096, 4000672, 4065248, 4128800, 4193376,
# /usr/bin/mkdir /vcd
# /usr/sbin/mount /dev/dsk/c0t0d0s7 /vcd
# /usr/bin/mkdir -m 755 /vcd/s0
#
```

6. Place the "Solaris™ 8 Software 1 of 2" CD into the CD drive.
   Start the Volume Manager (volmgt)

```
# /etc/init.d/volmgt start
#
```

7. Copy the contents of the CD slice s0 to the virtual drive.

```
# cd /cdrom/sol_8_401_sparc/s0
# /usr/bin/find . -print | cpio -pudm /vcd/s0
1037176 blocks
#
```

8. Stop the Volume Manager

```
# cd /vcd
# /etc/init.d/volmgt stop
#
```

9. The contents of slices 2 through 5 contain a single file ./SUNW-boot-redirect, which redirects the OpenBoot PROM (OBP) boot loader to load the kernal from slice 1. Elimination of slices for architectures other then sun4u (slice 5) is possible in order to allow for greater space for slice 0. This would require modification of the slice layout that was defined by the Software CD and can be done through the creation and writing of the dk1_vtoc and dk_label structures.

Table 1 outlines what slices are associated with what architectures:

Table 1 – CD Slices

0	• First 512 Bytes carries the VTOC of the CD-ROM • HSFS volume with the Solaris distribution
1	• The gerneric kernel • UFS system / directory after boot
2	Boot block for the sun4c architecture
3	Boot block for the sun4m architecture
4	Boot block for the sun4d architecture
5	Boot block for the sun4u architecture

10. Copy the required slices (s1 and s5) the VTOC, and restart the Volume Manager:

```
# for i in 1 2 3 4 5
> do
> dd if=/dev/dsk/c0t2d0s${i} of=/vcd/sol8.s${i} bs=512
> done
172800+0 records in
172800+0 records out
2560+0 records in
2560+0 records out
2560+0 records in
2560+0 records out
2560+0 records in
2560+0 records out
2560+0 records in
2560+0 records out
# /usr/bin/dd if=/dev/dsk/c0t2d0s0 of=/vcd/sol8.cdrom.vtoc bs=512
count=1
1+0 records in
1+0 records out
#
# /etc/init.d/volmgt start
volume management starting.
#
```

11. In order to create customized JumpStart™ profile and rules while reducing disk space requirements, the sample configuration files from /vcd/s0/.install_config will be removed. Delete the files under /vcd/s0/.install_config. Create a rule file and a profile with the Core Solaris distribution for a 64-bit architecture and the additional packages previously identified.

```
# cd /vcd/s0/.install_config
# /usr/bin/rm /vcd/s0/.install_config/*
# /usr/bin/vi /vcd/s0/.install_config/rules
  any -   -   basic.profile  -

# /usr/bin/vi /vcd/s0/.install_config/basic.profile
  install_type    initial_install
  system_type     standalone
  boot_device     c0t0d0s0 preserve
  usedisk         c0t0d0
  geo             N_America
  isa_bits        64
  partitioning    explicit
  filesys         rootdisk.s0  256       /     logging
  filesys         rootdisk.s1  2048     /var   logging,nosuid,noatime
  filesys         rootdisk.s3  2048     swap
  filesys         rootdisk.s4  1024     /usr    logging
  filesys         rootdisk.s5  1024     /opt    logging
  filesys         rootdisk.s6  free     /local  logging,nosuid
  filesys         rootdisk.s7  2048     /vcd    logging,nosuid
  cluster         SUNWCreq
  package         SUNWarc add
  package         SUNWarcx add
  package         SUNWast add
  package         SUNWbtoox add
  package         SUNWmkcd add
  package         SUNWdoc add
  package         SUNWesxu add
  package         SUNWxwfnt add
  package         SUNWgzip add
  package         SUNWzlib add
  package         SUNWzlibx add
  package         SUNWless add
  package         SUNWjvjit add
  package         SUNWjvrt add
  package         SUNWmfrun add
  package         SUNWntpr add
  package         SUNWntpu add
  package         SUNWxwice add
  package         SUNWxwplt add
  package         SUNWctplx add
  package         SUNWctpls add
  package         SUNWtoo add
  package         SUNWtoox add
  package         SUNWbtool add
  package         SUNWsprot add
  package         SUNWlibm add
  package         SUNWscpu add
  package         SUNWscpux add
  package         SUNWluxd add
  package         SUNWlibC add
  package         SUNWsprox add
  package         SUNWlibCx add
  package         SUNWhmdu add
  package         SUNWhea add
  package         SUNWlibCf add
  package         SUNWaccr add
  package         SUNWaccu add
  package         SUNWtnfc add
  package         SUNWtnfcx add
  package         SUNWtnfd add
  package         SUNWter add
  package         SUNWtltk add
  package         SUNWtltkx add
  package         SUNWvolr add
  package         SUNWvolu add
  package         SUNWvolux add
  package         SUNWxwrtl add
  package         SUNWxwicx add
  package         SUNWxwrtx add
  package         SUNWxwplx add
  package         SUNWxcu4 add
  package         SUNWxcu4x add
  package         SUNWxilow add
  package         SUNWxildh add
  package         SUNWxilrl add
  package         SUNWeuluf add
  package         SUNWeulux add
  package         SUNWm64x delete
```

12. Some of the required packages exist on the Solaris 8 Software CD 2 of 2. Fortunately, Sun created these drives to be combined in a JumpStart area and the packages are only split up because of the CD space restrictions. This means that the package configuration file for both CDs exist on CD 1 of 2. Copy over desired packages. If packages from locations other then the Solaris 8 Software CD 2 of 2 are copied over, update the .order and the .packagetoc files. To avoid problems with the CD disk space limitations, delete as much or more disk space as what is added. Eject the Solaris CD 1 of 2, and placed the Solaris disk 2 of 2.

```
# /usr/bin/eject cdrom
#
```

13. Check disk space and then copy the desire packages into the /vcd area.

```
# /usr/sbin/df -k /vcd
  /dev/dsk/c0t0d0s7    2052750  544848 1446320    28%    /vcd
# cd /cdrom/cdrom0/Solaris_8/Product
# for i in SUNWarc SUNWarcx SUNWast SUNWbtoox SUNWmkcd SUNWgzip \
  SUNWzlib SUNWzlibx SUNWless SUNWbtool SUNWsprot SUNWlibm  \
  SUNWscpux SUNWsprox SUNWhea SUNWaccr SUNWaccu SUNWtnfc SUNWtnfcx \
  SUNWtnfd SUNWter
  > do
  > /usr/bin/find ${i} -print | cpio -pudm /vcd2/s0/Solaris_8/Product
  > done
  4674 blocks
  1873 blocks
  112 blocks
  202 blocks
  104 blocks
  112 blocks
  142 blocks
  114 blocks
  158 blocks
  728 blocks
  1236 blocks
  247 blocks
  203 blocks
  18 blocks
  4872 blocks
  33 blocks
  246 blocks
  180 blocks
  185 blocks
  51 blocks
  1018 blocks
```

14. Now delete some, if not all, of the packages that are not going to be installed. Then make sure that disk space usage is equal to or less then when the packages were copied over.

```
# for i in SUNWi2of  SUNWhea SUNWcttf SUNWkxmft SUNWdtdst SUNW5xmft \
    SUNWilof SUNWkxfnt SUNWjmfp SUNWxwoft SUNWwbcou SUNWdthez
SUNWkttf \
    SUNWgxfnt  SUNWgttf SUNWjxmft SUNWxwcft SUNWhxfnt  SUNWjxcft  \
    SUNW5ttf SUNWdthev  NSCPcom
  > do
  > /usr/bin/rm -r ${i}
  > done

# /usr/bin/df -k /vcd
  /dev/dsk/c0t0d0s7    2052750  445564 1545604    23%    /vcd
```

15. On a trusted network, a JumpStart server can be created that would provide the sysidcfg information. The sysidcfg, rules, and profile files can also be put on a floppy disk. The system will first try the network and then it will read the floppy disk during boot up. Place a floppy disk into the drive. Create the sysidcfg file, prepare a floppy, and then copy the file onto the floppy disk.

```
# /usr/bin/vi /vcd/so/.install_config/sysidcfg
  name_service=DNS {domain_name=crobbins.org name_server=172.16.129.157}
  network_interface=hme0 {netmask=255.255.255.0 protocol_ipv6=no}
  security_policy=NONE
  timezone=US/Eastern
  timeserver=localhost
  terminal=sun-cmd

# /usr/bin/volcheck
# /usr/bin/fdformat -U
Formatting 1.44 MB in /dev/rdiskette
Press return to start formatting floppy.
.....................................................................
..
# /usr/sbin/newfs  /dev/rdiskette
/vol/dev/aliases/floppy0:       2880 sectors in 80 cylinders of 2 tracks, 18
sec
tors
        1.4MB in 5 cyl groups (16 c/g, 0.28MB/g, 128 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
 32, 640, 1184, 1792, 2336,

# /usr/bin/mkdir /floppy/scrap
# /usr/sbin/mount /dev/diskette /floppy/scrap
# /usr/bin/cat /etc/mnttab | grep "floppy"
/vol/dev/diskette0/unnamed_floppy       /floppy/scrap   ufs
rw,intr,largefil
es,onerror=panic,suid,dev=16c000d       995317950
# /usr/bin/cp /vcd/s0/.install_config/sysidcfg /floppy/scrap
#
```

16. Check the rules and profile files. The pfinstall command will complain about inconsistencies in the current state and state that the system must be rebooted. Ignore these message.

```
# cd /vcd/s0/.install_config
# /vcd/s0/Solaris_8/Misc/jumpstart_sample/check -r rules -p /vcd/s0
Validating rules...
Validating profile S2.8-Standard.profile...
The custom JumpStart configuration is ok.
# /vcd/s0/Solaris_8/Tools/Boot/usr/sbin/install.d/pfinstall -D -c /vcd/s0
basic.profile
#
```

17. Modify profind script which mounts the path to the JumpStart™ directory:

```
# /usr/bin/vi /vcd/s0/Solaris_8/Tools/Boot/usr/sbin/install.d/profind
#
```

Replace:

```
cdrom()
{
    # Factory JumpStart is only allowed with factory
    # stub images, indicated by the file /tmp/.preinstall
    #
    if [ -f /tmp/.preinstall ]; then
        mount -o ro -F lofs ${CD_CONFIG_DIR} \
                ${SI_CONFIG_DIR} >/dev/null 2>&1

        if [ $? -eq 0 ]; then
            verify_config "defaults" "CDROM"
        fi
    fi
}
```

with:

```
cdrom()
{
    gettext " <<< using CD Default >>>"; echo
    rmdir ${SI_CONFIG_DIR}
    ln -s /cdrom/.install_config ${SI_CONFIG_DIR}
    exit 0
}
```

18. If SUNWmkcd is not installed, put the Solaris 8 CD 2 of 2 into the CD-ROM drive and add the package.

```
# cd /cdrom/cdrom0/Solaris_8/Product
# /usr/sbin/pkgadd -d . SUNWmkcd
#
```

19. The package SUNWmkcd includes the command "mkisofs." Use mkisofs to convert the modified /vcd/s0 into a HSFS filesystem /vcd/sol8.s0:

```
# /usr/bin/mkisofs -R -d -L -l -o /vcd/sol8.s0 /vcd/s0
Total extents actually written = 279895
Total translation table size: 0
Total rockridge attributes bytes: 4329999
Total directory bytes: 24875008
Path table size(bytes): 178970
Max brk space used 16d8000
279895 extents written (546 Mb)
#
```

20. Copy the HSFS filesystem out to a single file and remove HSFS filesystem in order to clear off room:

```
# /usr/bin/dd if=/vcd/sol8.s0 of=/vcd/new.sol8.s0 bs=512 skip=1
1119579+0 records in
1119579+0 records out
# /usr/bin/rm /vcd/sol8.s0
```

21. In order to maintain the correct cylinder boundaries, the VTOC must be padded using the dd command with /dev/zero. The count is computed by subtracting from partition 0 sectors count of the unmodified VTOC (see results from step 3 prtvtoc command) the sum of one plus the number of sectors in the HSFS slice 0 (see results of step 18 dd command).

    Sector Count from prtvtoc from Step 3 = 1120000
    Sectors in the HSFS slice 0 from Step 17 = 1119579

    1120000 – (1119579 + 1) = 420

```
# /usr/bin/dd if=/dev/zero of=/vcd/pad.s0 bs=512 count=420
420+0 records in
420+0 records out
#
```

22. Concatenate everything into one image file. Note the file size. On Roo, if everything was done correctly , the size should be 667,156,480. If the image file size appears to be correct, write the image out to the CD-RW drive.

```
# cd /vcd
# /usr/bin/cat sol8.cdrom.vtoc new.sol8.s0 pad.s0 sol8.s1 sol8.s2
sol8.s3 \
  sol8.s4 sol8.s5 >cd8.image
# /usr/bin/ls -la /vcd/cd8.image
-rw-------  1 root    staff    667156480  Jul 11 19:44 /vcd/cd8.image
# /usr/bin/cdrw -d /dev/rdsk/c1t4d0s2 -i cd8.image
#
```

23. To use the newly created CD to do an install, place the CD in the CD-ROM drive of an installation client. Place the floppy in the installation client floppy drive. From the ok prompt boot off the CD-ROM drive.

```
ok boot cdrom - install
ok
```

Appendix

A.1 Packages Installed

Legend

1 – Solaris 8 04/01 Core Package Required
2 - Solaris 8 04/01 Core Package Checked
3 – Unchecked 32-bit, but 64-bit checked
4 – Unchecked 64-bit, but 32-bit checked
5 – Support for headers, libraries, and programming tools
6 – Network Time Protocol (NTP) support
7 – CD Creation Tools
8 – Support for OpenSSH and OpenSSH X Tunneling
9 – Support for GNU tools
10 – Client Graphic Support
11 – Terminal Information
12 – Accounting
13 – Berkeley Unix support
14 – Sunscreen
15 – Automated Security Enhancement Tools
* - Core Cluster Packages that can be removed

Packages from "Solaris 8 disk 1 of 2" CD

Audio Drivers (64-bit)	SUNWauddx	2 *
Audi drivers and applications		
Audio Applications	SUNWauda	1 *
Audio Drivers	SUNWaudd	1 *
AutoFS, (Root)	SUNWatfsr	1 *
AutoFS, (Usr)	SUNWatfsu	1 *
Core Architecture (Kvm) (64-bit)	SUNWkvmx	1
Core Architecture, (Kvm)	SUNWkvm	1
Core Architecture, (Root)	SUNWcar	1
Core Architecture, (Root) (64-bit)	SUNWcarx	2
Core Solaris		
Core Solaris Devices	SUNWcsd	1
Core Solaris, (Root)	SUNWcsr	1
Core Solaris, (Shared Libs)	SUNWcsl	1
Core Solaris, (Usr)	SUNWcsu	1
Sendmail root	SUNWsndmr	1
Sendmail usr	SUNWsndmu	1

Core Solaris (Usr) (64-bit)	SUNWcsxu	2
Core Solaris Libraries (64-bit)	SUNWcslx	2
Documentation Tools	SUNWdoc	14
Extended System Utilities	SUNWesu	1
Extended System Utilities (64-bit)	SUNWesxu	4
FTP Server, (Root)	SUNWftpr	1
FTP Server, (Usr)	SUNWftpu	1
Font Server Cluster		
X Window System platform required fonts	SUNWxwfnt	10
Framebuffer Device Drivers		
Dumb Frame Buffer Device Drivers	SUNWdfb	1
GX (cg6) Device Driver (64-bit)	SUNWcg6x	2
GX (cg6) OS Support Files		
GX (cg6) Device Driver	SUNWcg6	1
Install and Patch Utilities	SUNWswmt	1
JavaVM		
Java JIT compiler	SUNWjvjit	14
JavaVM run time environment	SUNWjvrt	14
Keyboard configuration tables	SUNWkey	1
M64 Graphics Accelerator Support		
M64 Graphics System Software/Device Driver	SUNWm64	1 *
M64 Graphics Accelerator Support (64-bit)		
M64 Graphics System Software/Device Driver (64-bit)	SUNWm64x	1 *
Motif RunTime Kit	SUNWmfrun	10
Network Information System (NIS)		
Network Information System, (Root)	SUNWnisr	1 *
Network Information System, (Usr)	SUNWnisu	1 *
Network Time Protocol		
NTP, (Root)	SUNWntpr	6
NTP, (Usr)	SUNWntpu	6
OpenWindows Version 3		
ICE components	SUNWxwice	10
OpenWindows kernel modules	SUNWxwmod	1
X Window System platform software	SUNWxwplt	10
X Windows System Window Drivers	SUNWxwdv	1
OpenWindows Version 3 (64-bit)		
X Window System kernel modules (64-bit)	SUNWxmox	2
X Windows System Window Driver (64-bit)	SUNWxwdvx	2
PCI Drivers		
PCI Drivers	SUNWpd	1
PCI Drivers (64-bit)	SUNWpdx	2
PCMCIA support		

3COM EthernetLink III PCMCIA Ethernet Driver	SUNWpcelx	1 *
PCMCIA ATA card driver	SUNWpsdpr	1 *
PCMCIA Card Services, (Root)	SUNWpcmci	1 *
PCMCIA Card Services, (Usr)	SUNWpcmcu	1 *
PCMCIA memory card driver	SUNWpcmem	1 *
PCMCIA serial card driver	SUNWpcser	1 *
PCMCIA support (64-bit)		
PCMCIA Card Services (64-bit)	SUNWpcmcx	2 *
Perl 5		
Perl 5.005_03	SUNWpl5u	1 *
Portable layout services for CTL (64-bit)	SUNWctplx	10
Portable layout services for Complex Text Layout support	SUNWctpls	10
Programming Tools	SUNWtoo	5
Programming Tools (64-bit)	SUNWtoox	5
Programming tools and libraries		
Sun Workshop Bundled shared libm	SUNWlibms	1
Realmode Modules, (Usr)	SUNWrmodu	1 *
SCSI Enclosure Services Device Drivers	SUNWses	1 *
SCSI Enclosure Services Device Driver (64-bit)	SUNWsesx	2 *
SPARCstorage Array Drivers	SUNWssad	1 *
SPARCstorage Array Drivers (64-bit)	SUNWssadx	2 *
Solaris Desktop /usr/dt filesystem anchor	SUNWdtcor	1
Solaris Naming Enabler	SUNWsolnm	1 *
Solaris Product Registry & Web Start runtime support	SUNWwsr2	1 *
Source compatibility support		
Source Compatibility, (Usr)	SUNWscpu	13
Sun Enterprise Network Array Drivers & Utilities		
Sun Enterprise Network Array sf Device Driver	SUNWluxd	3 *
Sun Enterprise Network Array firmware and utilities	SUNWluxop	1 *
Sun Enterprise Network Array libraries (64-bit)	SUNWluxox	2 *
Sun Enterprise Network Array sf Device Driver (64-bit)	SUNWluxdx	2 *
Sun Fibre Channel Transport Software		
Sun FCIP IP/ARP over FibreChannel Device Driver	SUNWfcip	1 *
Sun FCP SCSI Device Driver	SUNWfcp	1 *
Sun Fibre Channel Transport layer	SUNWfctl	1 *
Sun Fibre Channel Transport Software (64-bit)		
Sun FCIP IP/ARP over FibreChannel Device Driver (64-bit)	SUNWfcipx	2 *
Sun FCP SCSI Device Driver (64-bit)	SUNWfcpx	2 *
Sun Fibre Channel Transport layer (64-bit)	SUNWfctlx	2 *
Sun Quad FastEthernet Adapter Driver (64-bit)	SUNWqfedx	2
Sun Quad FastEthernet PCI/SBus Adapter Software		
Sun Quad FastEthernet Adapter Driver	SUNWqfed	1
Sun RIO 10/100 Mb Ethernet Drivers (64-bit)	SUNWeridx	2
Sun Workshop Bundled 64-bit libC	SUNWlibCx	5
Sun Workshop Bundled 64-bit shared libm	SUNWlmsx	2

Sun Workshop Compilers Bundled libC	SUNWlibC	5
SunFastEthernet/FastWideSCSI-2 Adapter Drivers		
SunSwift SBus Adapter Drivers	SUNWhmd	1
SunSwift SBus Adaper Header	SUNWhmdu	5
SunSoft Workshop Bundled libC (cfront version)	SUNWlibCf	5
SunSwift SBus Adapter Drivers (64-bit)	SUNWhmdx	2
System Localization	SUNWloc	1
System Localization (64-bit)	SUNWlocx	2
System and Network Admin		
System & Network Administration Root	SUNWadmr	1 *
ToolTalk End User		
ToolTalk runtime	SUNWtltk	10
ToolTalk library (64-bit)	SUNWtltkx	10
USB Device Drivers (64-bit)	SUNWusbx	2 *
Universal Disk Format 1.50 (64-bit)	SUNWudfrx	2
Universal Disk Format File System		
Universal Disk Format 1.50	SUNWudfr	1
Universal Disk Format 1.50, (Usr)	SUNWudf	1
Universal Serial Bus Software		
USB Device Drivers	SUNWusb	2 *
Volume Management		
Volume Management, (Root)	SUNWvolr	14
Volume Management, (Usr)	SUNWvolu	14
Volume Management (64-bit)		
Volume Management (Usr) (64-bit)	SUNWvolux	14
X Window System & Graphic Runtime Library in /usr/lib	SUNWxwrtl	8
X Window System ICE library (64-bit)	SUNWxwicx	8
X Window System Runtime Compatibility Package (64-bit)	SUNWxwrtx	8
X Window System library software (64-bit)	SUNWxwplx	8
XCU4 Utilities	SUNWxcu4	8
XCU4 Utilities (64-bit)	SUNWxcu4x	8
XIL Runtime Environment		
XIL Desktop Loadable Pipeline Libraries	SUNWxilow	10
XIL Loadable Pipeline Libraries	SUNWxildh	10
XIL Runtime Environment	SUNxilrl	10
en_US.UTF-8		
UTF-8 L10N For Language Environment User Files	SUNWeuluf	14
en_US.UTF-8 (64-bit)		
UTF-8 L10N For Language Environment User Files (64-bit)	SUNWeulux	14

Packages from “Solaris 8 disk 2 of 2” CD

Archive Libraries	SUNWarc	5
Archive Libraries (64-bit)	SUNWarcx	5
Automated Security Enhancement Tools	SUNWast	15

CCS libraries bundled with SunOS (64-bit)	SUNWbtoox	4
CD Creation utilities	SUNWmkcd	7
Freeware Compression Utilities		
The GNU Zip (gzip) compression utility	SUNWgzip	9
The Zip compression library	SUNWzlib	8
Freeware Compression Utilities (64-bit)		
The Info-Zip compression library (64-bit)	SUNWzlibx	8
Freeware Other Utilities		
The GNU pager (less)	SUNWless	9
Programming tools and libraries		
CCS tools bundled with SunOS	SUNWbtool	5
Solaris Bundled tools	SUNWsprot	5
Sun Workshop Bundled libm	SUNWlibm	5
Source compatibility support (64-bit)		
Source Compatibility (Usr) (64-bit)	SUNWscpux	13
Sun Workshop Bundled 64-bit make library	SUNWsprox	14
SunOS Header Files	SUNWhea	5
System Accounting		
System Accounting, (Root)	SUNWaccr	12
System Accounting, (Usr)	SUNWaccu	12
TNF Core Components	SUNWtnfc	5
TNF Core Components (64-bit)	SUNWtnfcx	5
TNF Developer Components	SUNWtnfd	5
Terminal Information	SUNWter	11

Packages from the "Software Supplemental for the Solaris 8 Operating System" CD

CD read and write utility for Solaris	SUNWcdrw	7

A.2 Trace.pl and Sample Output

```
  #!/usr/local/bin/perl

  # Generated with truss -f -o /var/tmp/truss.out -t open -t execve -t start -t access -t
creat /usr/local/apache/bin/apachectl startssl

  # Assumed location of truss.out
  my $infile = "/var/tmp/truss.out";
  my $cfile = "/var/sadm/install/contents";

  # For sorting
  my %pak;
  my %check;
  my %reqp;

  # Place the contents file into an array for easy access
  local($/,$*) = ("\n",1);
  open(INFILE,$cfile);
  while (my $line = <INFILE>) {

     # First check if file is refernecs, then check if it is a
     # a link.
     if ($line =~ m/^([^\s\=]+)[^\=]*\s+(\S+)$/) {
        my ($cpak,$clib) = ($1,$2);
        $pak{$cpak}=$clib;
     }
     elsif ($line =~ m/^([^\s\=]+)\=(\S+).*\s+(\S+)$/) {
        my($cpak,$link,$clib)=($1,$2,$3);
        $pak{$cpak}=$clib;

        # Follow the link and track both
        if ($link =~ m/^\.\.\/(.*)$/) {
           my $rep = $1;
           if ($cpak =~ m/^(.*\/)[^\/]+\/[^\/]+$/) {
              my $rem = $1 . $rep;
              $pak{$rem}=$clib;
           }
           else {
              print("CONFUSED: ($link) ($cpak)\n");
           }
        }
        elsif ($link =~ m/^\.\/(.*)$/) {
          my $rep = $1;
          if ($cpak =~ m/^(.*\/)[^\/]+$/) {
              my $rem = $1 . $rep;
              $pak{$rem}=$clib;
           }
           else {
              print("CONFUSED: ($link) ($cpak)\n");
           }
        }

     }
  }
  close(INFILE);

  # Examine contents of truss output file for valid access to existing
  # files
  open(INFILE,$infile);
  while (my $line = <INFILE>) {

    if (($line =~ m/^\d+:\s+\w+\(\"([^\"]+)\"/) && ($line !~ /Err#2 ENOENT$/))  {
       my $opfile = $1;
       if (! (-d $opfile)) {
         $check{$opfile}++;
       }
    }
    else {
  #     print("PROBLEM: parsing line ($line)\n");
    }
  }
  close(INFILE);
```

```
# Consolidate packages with files accessed under that package
foreach my $key (sort keys %check) {
   if (defined $pak{$key}) {
      $reqp{$pak{$key}} .= "   $key" . "\n";
   }
}

# Print results
foreach my $key (sort keys %reqp) {
   my $results = `/usr/bin/pkginfo $key`;
   if ($results =~ m/^\S+\s+\S+\s+(.*)$/) {
       my $des = $1;
       print "$key ($des) required by:\n" . $reqp{$key} . "\n";
   }
   else {
      print("odd results ($results)\n");
   }
}
```

A.3 SunScreen Configurations

```
# /opt/SUNWicg/SunScreen/bin/ssadm edit Initial
Loaded common objects from Registry version 10
Loaded policy from Initial version 16
edit> list addresses
"kanga" HOST 172.16.129.52
"pooh" HOST 172.16.129.157
"eeyore" HOST 172.16.129.166
"hme0.net" RANGE 172.16.129.0 172.16.129.255
"tigger" HOST 172.16.129.138
"roo_hme0" GROUP { } { }
"syslog-clients" GROUP { "kanga" "eeyore" "tigger" } { }

edit>  list rules
1 "ssh" "localhost" "kanga" ALLOW
2 "syslog" "syslog-clients" "localhost" ALLOW
3 "ntp" "localhost" "pooh" ALLOW
4 "dns" "localhost" "pooh" ALLOW
5 "ping" "localhost" "syslog-clients" ALLOW COMMENT "Check clients are up"
6 "ssh" "localhost" "syslog-clients" ALLOW COMMENT "Check ssh clients"

edit> list services
"ah" SINGLE FORWARD "iptunnel" PORT 51
"archie" SINGLE FORWARD "udp" PORT 1525 PARAMETERS  360 -1 0
"auth" SINGLE FORWARD "tcp" PORT 113
"automount" SINGLE FORWARD "pmap_tcp" PORT 300019 FORWARD "pmap_udp" PORT
300019 FORWARD "rpc_tcp" PORT 300019 FORWARD "rpc_udp" PORT 300019
"Backweb" SINGLE FORWARD "udp" PORT 370 PARAMETERS  60 0 3
"biff" SINGLE FORWARD "udp_datagram" BROADCAST 512
"bootp" SINGLE FORWARD "udp" BROADCAST 67 PARAMETERS  60 0 3
"certificate discovery" SINGLE FORWARD "udp" PORT 1640 PARAMETERS  60 1 1
"chargen" SINGLE FORWARD "tcp" PORT 19
"common" GROUP "tcp all" "udp all" "syslog" "dns" "rpc all" "nfs prog" "icmp all" "rip" "ftp"
"rsh"
"real audio" "pmap udp all" "pmap tcp all" "rpc tcp all" "nis" "archie" "traceroute" "ping"
"common services" GROUP "tcp all" "udp all" "syslog" "dns" "rpc all" "nfs prog" "icmp all" "rip"
"ftp"
"rsh" "real audio" "pmap udp all" "pmap tcp all" "rpc tcp all" "nis" "archie" "traceroute" "ping"
"CoolTalk" SINGLE FORWARD "tcp" PORT 6499-6500 FORWARD "udp_datagram" PORT 13000
REVERSE "udp_datagram" PORT 13000
"CU See Me" SINGLE FORWARD "udp_datagram" PORT 7648-7652
"daytime" SINGLE FORWARD "tcp" PORT 13
"daytime group" GROUP "daytime" "daytime-udp"
"daytime-udp" SINGLE FORWARD "udp" PORT 13
"discard" SINGLE FORWARD "tcp" PORT 9
"discard group" GROUP "discard" "discard-udp"
"discard-udp" SINGLE FORWARD "udp" PORT 9
"dns" SINGLE FORWARD "tcp" PORT 53 FORWARD "dns" PORT 53
"echo" SINGLE FORWARD "tcp" PORT 7
"echo group" GROUP "echo" "echo-udp"
"echo-udp" SINGLE FORWARD "udp" PORT 7
"esp" SINGLE FORWARD "iptunnel" PORT 50
"exec" SINGLE FORWARD "tcp" PORT 512
"finger" SINGLE FORWARD "tcp" PORT 79
"ftp" SINGLE FORWARD "ftp" PORT 21
"gopher" SINGLE FORWARD "tcp" PORT 70
"HA" GROUP "HA heartbeat" "HA administration"
"HA administration" SINGLE FORWARD "tcp" PORT 3853
"HA heartbeat" SINGLE FORWARD "ping" PORT 8
"icmp all" SINGLE FORWARD "icmp" PORT * BROADCAST *
"icmp echo-reply" SINGLE FORWARD "icmp" PORT 0
"icmp echo-request" SINGLE FORWARD "icmp" PORT 8
"icmp exceeded" SINGLE FORWARD "icmp" PORT 11
"icmp info" SINGLE FORWARD "icmp" PORT 13 PORT 14 PORT 15 PORT 16 PORT 17 PORT 18
"icmp params" SINGLE FORWARD "icmp" PORT 12
"icmp quench" SINGLE FORWARD "icmp" PORT 4
"icmp redirect" SINGLE FORWARD "icmp" PORT 5
"icmp unreach" SINGLE FORWARD "icmp" PORT 3
"imap" SINGLE FORWARD "tcp" PORT 143
"ip all" SINGLE FORWARD "ip" PORT *
"ip forward" SINGLE FORWARD "ipfwd" PORT *
"ip mobile" SINGLE FORWARD "ipmobile" PORT *
"ipsec" GROUP "esp" "ah" "isakmp"
"ip tunnel" SINGLE FORWARD "iptunnel" PORT *
"ipv6 tunnel" SINGLE FORWARD "iptunnel" PORT 41
"irc" SINGLE FORWARD "tcp" PORT 6670 FORWARD "tcp" PORT 6680
"isakmp" SINGLE FORWARD "udp" PORT 500
"kerberos" SINGLE FORWARD "udp" PORT 88
"lpd" SINGLE FORWARD "tcp" PORT 2766
"mosaic" GROUP "www" "ssl" "gopher" "ftp"   "archie"
"mountd" SINGLE FORWARD "rpc_tcp" PORT      100005 FORWARD "rpc_udp" PORT 100005
FORWARD "pmap_tcp" PORT 100005 FORWARD      "pmap_udp" PORT 100005
"netbios" GROUP "netbios name" "netbios datagram" "netbios session"
"netbios datagram" SINGLE FORWARD "udp_datagram" PORT 138 BROADCAST 138
```

A.4 Program Listing for mirror_boot.sh[7]

```
#!/bin/sh
#
# SCRIPT:   /secure/mirror_boot.sh
#
# FUNCTION: backup main boot disk to secondary (cold mirroring).
#           Each night (for example) the offline disk is mounted and
#           synchronised with the primary disk.
#           ***** See the latest doc and version of this script at: *****
#           http://securityportal.com/articles/coldmirroring20010304.html
#
# This script is typically called from the root cron nightly. It mounts the
# spare disk under /newroot, copies all filesystems, installs a boot block and
# copies over a new vfstab. This creates a fully updated bootable spare disk.
# The results of the script are sent to the administrator via email.
#
# USAGE:
#   Set the admin, mounts and targets variables below, then run via cron
#   when disk activity is as low as possible:
#   0 23 * * *   /secure/mirror_boot.sh
#   Set DEBUG to 1, to "dry run" the script without actually copying any data.
#
# ASSUMPTIONS:
#   - a second identical disk with identical partitions and filesystems exists.
#   - The /newroot mount point exists, e.g.:
#       mkdir /newroot; chmod 777 /newroot ; mount /newroot
#   - entries in vfstab exist for relevant filesystems (e.g. /newroot/usr
#     /newroot/opt /newroot/var)
#   - /etc/vfstab.newroot exists with correct device entries so that
#     we can boot from the second disk.
#   - The variables have been set below
#   - The /newroot partitions are unmounted when running this script.
#
# PROBLEMS: The /newroot target is never wiped clean, this has the advantage
# that files deleted several days ago can be recovered, but the disadvantage
# that the new targets are likely to fill up over time (as hence need wiping
# every few months or so). ufsdump could be used rather than cpio to copy the
# device, but then files more than one day old cannot be recovered.
#
# Solaris Intel is a bit tricky. See the notes at the bottom of this file.
#
# HISTORY:
# <7> 26.Feb.01 sb $BOOTABLE, $ignore_these, print eeprom boot-device
# <6> 19.12.00 sb Correct quoting, and minor fix for first time run. Ignore sockets.
# <5> 12.10.00 sb Shutdown mysql (or other app) before backup, restart after
#                 Don't abort if cpio has errors, continue as much as possible.
#                 Fixed 2>1 =>  2>&1 bugs
# <4> 6.10.00 sb ACL backups not working correctly, disable.
# <3> 25.9.00 sb Log result to syslog, add boot support for Intel Architecture.
#               tested on Solaris8 x86. cpio: backup ACLs & comment.
# <2> 20.7.00 sb /proc & mount point fixes. Auto get boot raw device.
#               Add VERBOSE variable to supress "ok" emails if wanted.
#               Tested on Solaris 2.8 too.
# <1> 16.6.00 sb Additional comments, add DEBUG+newroot_device variables
# <0> 28.8.99 sb Original script by Sean Boran on Solaris 2.7
#
# LICENSE:
#      This script was developed by Sean Boran, http://www.boran.com.
#      It can be distributed for free as long as these headers are included.
#      Please send any bug fixes or improvements to sean@boran.com.

###### Set the following variables according to your needs ############

# Dry run: show lots of detail on what would be done, but don't copy
DEBUG='1';

## Where should we email results?
admin='root';

## / will always be backed up, only list other filesystems
targets='/var /opt /usr /local'
#targets='/disk2'
```

```
## List empty mount points needed, we need /proc, local filesystems mounts,
## or remote NFS mounts. /var/run is also needed on Solaris8 <3>.
## oldroot is needed for mounting the primary disk, when the secondary mirror
## is used to boot the machine.
mounts='/proc  /var /var/run /opt /usr /local /oldroot';
#mounts='/proc /disk2 /var/run /oldroot';

## Send an email even if backup was OK?
VERBOSE='0';

## If you want specific applications stopped before the backup
## and restart afterwards, add in the commands here. <5>
## If there are no app daemons:
stop_daemons='';
start_daemons='';
## or a SecurID ACE Server:
#stop_daemons='sh /etc/rc3.d/MM100ace stop';
#start_daemons='sh /etc/rc3.d/MM100ace start';
## or MySQL:
#stop_daemons='sh /etc/rc3.d/S99msql stop';
#start_daemons='sh /etc/rc3.d/S99msql start';

## Is the primary disk bootable? Set to 0 if you don't want a bootblock
## installed
BOOTABLE='1';

## Don't backup files that match this (egrep) pattern
ignore_these='.tmp|^core';
# If we want to ignore tar archives:
#ignore_these='.tmp|^core|.gz|.Z';

#####################################################

## -- variables --
f=/var/tmp/f$$;
# assume backup doesn't work, until we get to the end.
PROBLEM='1';

# Cpio (file copy) options:
# - copy all files listed in stdin, create dirs, maintain dates,
#   overwrite new, preserve Solaris ACLs.
# - If debugging, add '-v' for verbose, but the output will be huge.
# - The option preserve Solaris ACLs '-P' kept giving errors like: <4>
#"Error with acl() of "usr/dt/dthelp/nls/en_US.UTF-8", errno 2, No such file or directory"
#cpio='cpio -pdmuP';
# - to exclude certain file patterns add '-f PATTERN'
# - to skip corrupted files use '-k' ??
cpio='cpio -pdmu';

##---------- functions -------
check_err () {
  ## Check result of last operation. If it was an error, abort the
  ## script, clean up and email results
  if [ "$*" != "0" ] ; then
    echo "SCRIPT $0 ABORTED: error." >>$f 2>&1
    umount_filesystems;
    umount /newroot;
    send_results;
    exit 1;
  fi
}

echo_f () {
  echo "$*" >>$f
}

send_results () {
  echo_f "  "
  echo_f "This email was generated by `uname -n`:$0 "

  ## Restart daemons, if any, <5>
  if [ "$start_daemons" != "" ] ; then
     echo_f "Restart key applications after backup.."
     echo_f "$start_daemons"
```

```
    $start_daemons  >>$f
  fi

  # In debug mode print to stdout, else email results
  if [ "$DEBUG" = "1" ] ; then
    cat $f
  else
    # <3> Log result to syslog, and only delete log if successful.
    if [ "$PROBLEM" = "1" ] ; then
      mailx -s "`uname -n` Error: Boot disk backup" $admin < $f
      logger -p daemon.alert "Error: Boot disk backup see $f"
    elif [ "$VERBOSE" = "1" ] ; then
      # <2>
      mailx -s "`uname -n` OK: Boot disk backup" $admin < $f 2>&1
      logger -p daemon.info "Boot disk backup OK"
      rm $f
    fi
  fi
}

make_mount_points () {
  echo "make mount points $mounts .."      >>$f
  for mnt in $mounts ; do
    if [ "$DEBUG" = "1" ] ; then
      echo "mkdir -p /newroot$mnt"
    else
      mkdir /newroot/$mnt >/dev/null 2>&1;
      chmod 777 /newroot/$mnt >/dev/null 2>&1;
    fi
  done
}

umount_filesystems () {
  echo "umount /newroot/{$targets} " >>$f
  for filesys in $targets ; do
    umount /newroot$filesys          >> $f 2>&1
  done
}

##---------- main -------
echo "----- Backup $targets to /newroot----"    >>$f
echo "Logfile: $f on `date`" >>$f
echo "  " >>$f
if [ "$DEBUG" = "1" ] ; then echo "DEBUG mode on"; fi
cd

## Stop daemons, if required <5>
if [ "$stop_daemons" !=  "" ] ; then
   echo "Stopping key applications before backup.." >>$f
   $stop_daemons  >>$f
fi

## First do Root
umount /newroot >> /dev/null 2>&1
mount /newroot >> $f 2>&1
check_err "$?";

echo "Backing up root filesystem at `date` to /newroot" >>$f
if [ "$DEBUG" = "1" ] ; then
  echo "find . -xdev -print | $cpio /newroot at `date`" >>$f
  echo "DEBUG MODE: no data actually copied" >> $f 2>&1
else
  #find . -xdev -print | $cpio /newroot  >> $f 2>&1
  # Ignore sockets, it causes grief <6>. Ignore specific files too.
  find . ! -type s -xdev -print| egrep -v "$ignore_these"| $cpio /newroot  >> $f 2>&1
  check_err "$?";
fi
make_mount_points;

## Then other filesystems
for filesys in $targets ; do
  echo "mounting /newroot$filesys" >> $f 2>&1
  mount /newroot$filesys >> $f 2>&1
  check_err "$?";
  cd $filesys; >> $f 2>&1
```

```
  check_err "$?";
  echo "Backing up $filesys at `date`" >>$f
  if [ "$DEBUG" = "1" ] ; then
    echo "find . -xdev -print | $cpio /newroot$filesys at `date`" >>$f
    echo "DEBUG MODE: no data actually copied" >> $f 2>&1
  else
    # Ignore sockets, it causes grief <6>. Ignore specific files too.
    find . ! -type s -xdev -print| egrep -v "$ignore_these"| $cpio /newroot$filesys  >> $f 2>&1
    ## cpio can give weird errors like the following, ignore return code for now:
    ## Error with lstat() of "grou ", errno 2, No such file or directory <5>
    check_err "$?";
  fi
done

echo " "     >>$f
df -k -F ufs >>$f 2>&1
echo " "     >>$f
echo "Prom boot order is: `eeprom boot-device` "     >>$f
echo " "     >>$f

echo "Copy over /etc/vfstab.newroot.... " >>$f
if [ ! -d /newroot/etc ] ; then
  # The first time we run this script, or newroot is empty <6>
  # so create it with Default Solaris7/8 settings
  mkdir /newroot/etc; chmod 755 /newroot/etc;
  chown root:sys /newroot/etc;
fi
mv /newroot/etc/vfstab         /newroot/etc/vfstab.$$ >/dev/null 2>&1
# Ignore errors on the last line, if may happen the first run.
cp /etc/vfstab.newroot         /newroot/etc/vfstab >> $f 2>&1
check_err "$?";

umount_filesystems;

# <2> Before we unmount /newroot, do a df and get device name
#echo "How about boot= `df -k -F ufs | grep newroot | awk '{print $1}'`" >>$f
newroot_device=`df -k -F ufs | grep newroot | awk '{print $1}' |sed 's/dsk/rdsk/'`
umount /newroot                          >> $f 2>&1
check_err "$?";

if [ "$BOOTABLE" = "1" ] ; then
  # Boot block handling is different for x86 and sparc <3>
  if [ `uname -m` = "i86pc" ] ; then
    # Solaris Intel uses slice 2 for boot and has different arguments
    x86_device=`echo $newroot_device|sed 's/s0/s2/'`;
    echo "Install boot block on (PC device) $x86_device " >>$f
    /usr/sbin/installboot /usr/platform/`uname -i`/lib/fs/ufs/pboot /usr/platform/`uname -
i`/lib/fs/ufs/bootblk  $x86_device;
    check_err "$?";

  else
    # assume sparc
    echo "Install boot block on $newroot_device " >>$f
    /usr/sbin/installboot /usr/platform/`uname -i`/lib/fs/ufs/bootblk $newroot_device;
    check_err "$?";
  fi
fi

echo " " >>$f
echo "Finished at `date` " >>$f
# No errors so far, so tell this to user
PROBLEM='0';

send_results;
############ end of script ##############


####################  Solaris Intel Notes ##################
#
# Solaris Intel: Booting from 2nd (backup) Disk <3>
#
# Example: "server1" has two scsi disks, the main one has target 0, the second
#  (target 1) is mirrored each night from the first. If the main disk fails,
#  proceed as follows to boot from the second disk:
#
```

```
# a) Stop the PC, Switch it on allow the bios to do it's checks.
# b) When the scsi controller is checking the disks, Press Ctrl-A (for example)
#    to enter the scsi menu.
# c) Change the boot device to scsi ID 1 (which is the ID of the second disk).
#    Save changes, and exit, and it will probably insist on a reboot.
# d) After scsi disk check, the "SunOS Secondary Boot" will start, press ESC to
#    enter it's menu and F2 "continue" about three times until the list of
#    boot disks is shown. Select the backup disk (target 1) and press continue,
#    then allow the boot process to continue.
# e) login as usual, check with "df" that you really are using the backup disk.
# f) Disable the mirroring script in the root cron, until the primary disk is
#    working again!

############################################################
```

References

[1] Brotzman, Lee and Pomeranz, Hal. "6.5 Linux/Solaris Practicum." SANS 2001. Baltimore, MD. May 2001.

[2] Sun Microsystems. "System Administration Guide, Volume 2." Palo Alto, CA. February 2000. URL: http://192.18.99.138/805-7229/805-7229.pdf (21 May 2001).

[3] Noordergraaf, Alex. "Solaris™Operating Environment Minimization for Security: A Simple, Reproducible and Secure Application installation Methodology Updated for Solaris 8 Operating Environment." Sun Blueprints™ Online. November 2000. URL: http://www.sun.com/blueprints/1100/minimize-updt1.pdf (5 June 2001).

[4] Sun Microsystems. "Solaris 8 (SPARC Platform Edition) Installation Guide." Palo Alto, CA. February 2000. URL: http://192.18.99.138/806-0955/806-0955.pdf (21 May 2001).

[5] Noordergraaf, Alex and Watson, Keith. "Solaris™ Operating Environment Security – Updated for Solaris 8 Operating Environment." Sun Blueprints™ Online. April 2001. URL: http://www.sun.com/blueprints/0401/security-updt1.pdf (5 June 2001).

[6] Sun Microsystems. "OpenBoot 3.x Command Reference Manual." Palo Alto, CA. February 2000. URL: http://192.18.99.138/806-1377-10/806-1377-10.pdf (21 May 2001).

[7] Sun Microsystems. "Solaris 8Advance Installation Guide." Palo Alto, CA. February 2000. URL: http://192.18.99.138/806-0957/806-0957.pdf (21 May 2001).

[8] Pomeranz, Hal, editor. "Solaris Security Step by Step Version 2.0." The SANS Institute. 2001.

[9] Pomeranz, Hal. "Configurator Toolkit." Deer Run Associates. 14 May 2001. URL: http://www.deer-run.com/~hal/jumpstart/configurator/ (18 July 2001).

[10] Reid, Jason and Watson, Keith. "Building and Deploying OpenSSH for the Solaris™ Operating Environment." Sun Blueprints™ Online. July 2001. URL: http://www.sun.com/blueprints/0701/openSSH.html (15 July 2001).

[11] Osser, William. "Auditing in the Solaris™ 8 Operating Environment." Sun Blueprints ™ Online. February 2001. URL: http://www.sun.com/blueprints/0201/audit_config.pdf (14 July 2001).

[12] Sun Microsystems. "SunScreen 3.1 Lite Installation Guide." Palo Alto, CA. June 2000.

URL:
http://docs.sun.com:80/ab2/coll.557.2/SSCRNLITEINST/@Ab2TocView/3448?Ab2Lang=C&Ab2Enc=iso-8859-1 (June 12 2001).

[13] Boran, Seán. "Hardening Solaris: Secure Installation of Basiton Hosts." 26 June 2001.
URL: http://www.boran.com/security/sp/Solaris_hardening3.html (27 June 2001).

[14] Stokely, Celeste. "Celeste's Tutorial On Solaris 2.x Modems & Terminals." Stokely Consulting. 01 June 2001.
URL: http://www.stokely.com/unix.serial.port.resources/modem.html (14 July 2001).

[15] Boran, Seán. "A Tool for Cold Mirroring of Solaris System Disks." SecurityPortal. 06 March 2001.
URL: http://www.securityportal.com/articles/coldmirroring20010306.printerfriendly.html (18 July 2001).

[16] Noordergraaf, Alex. "Building a JumpStart™ Infrastrucure." Sun BluePrints™ Online. April 2001. URL: http://www.sun.com/blueprints/0401/BuildInf.pdf (12 June 2001).

[17] Howard, John S. "Building a Bootable JumpStart™ Installation CD-ROM." Sun BlePrints™ Online. March 2001.
URL: http://www.sun.com/blueprints/0401/BuildBoot.pdf. (12 June 2001).