Global Information Assurance Certification Paper

GIAC Certified UNIX Security Administrator (GCUX) Practical Assignment
Version 1.9

Securing a DNS server running BIND 9.2.2rc1 on Solaris 8

Submitted By:

John Worthing

Date Submitted:

15 October 2002

Table of Contents

Description of the system

The server in question is a Sun Ultra 5 (UltraSPARC-IIi 360MHz) with 512 MB memory and an 8 GB hard drive. There will be no external SCSI devices attached to this system nor will any of the available PCI slots be utilized. Only one ethernet interface is needed for this server and the onboard hme interface will do fine.

When complete, this system will be a secure, secondary DNS server built to run at a co-location facility on the east coast. Solaris 8 will be installed on the system, and then it will be patched and hardened. In the end, this server will run BIND 9.2.2rc1 in a chroot jail and assist in answering DNS queries for our domains more efficiently in that region of the US.

- **Risk analysis of the system**

Although this system will be behind a firewall, the server will still be assigned a routable, public IP address by the co-location facility to simplify administration (mainly NAT). Since this firewall is not administered by our group and the network this server is being placed in is basically their DMZ, fully securing and hardening this box is a top priority.

Physical access to the box is extremely limited since the server is kept in a locked cabinet. In addition to this server, a small portmaster also resides in this cabinet that provides console access to this machine. This device is owned and administered by our group as well.

User access to this machine will be very limited – three accounts will be created for each of the administrators that may need to admin this box. Root access will be disabled through both the console and ssh, and sudo will be installed and configured to log an audit trail.

The server only needs to run two internet services: domain udp/tcp on port 53 and ssh on port 22. Port 53 is the udp port BIND receives and answers DNS queries on and is also the tcp port named-xfer communicates on. Port 22 is the ssh port and will be used for remote administration. All other ports can be closed by disabling services, shutting down daemons, or reconfiguring certain applications.

So the key security objectives/concerns are

- patching and hardening the operating system to its fullest;
- BIND or BIND-related exploits that can result in either denial of service or buffer overflow conditions – so patching and hardening BIND to its fullest; and
- similar exploits that effect OpenSSH, or the applications used to build OpenSSH, which means fully patching and hardening OpenSSH, OpenSSL, and zlib.

- **Step by step guide to securing this server**

Step 1 – Install the OS

Insert the Solaris 8 CD (07/01 release) and boot to cdrom.

1.1 System configuration and identification
- 1.1.1 Networked: Yes
- 1.1.2 Use dhcp: No
- 1.1.3 Primary interface: hme0
- 1.1.4 Hostname: ns2
- 1.1.5 IP address: 10.1.1.9
- 1.1.6 Part of a subnet: Yes
- 1.1.7 Netmask: 255.255.255.0
- 1.1.8 Enable Ipv6: No
- 1.1.9 Configure Kerberos: No
- 1.1.10 Name service: None
- 1.1.11 Geographic region: United States
- 1.1.12 Time Zone: Eastern
- 1.1.13 Date: verify date
- 1.1.14 Time: verify time
- 1.1.15 Core System Support Installation
- 1.1.16 Choose Initial install
- 1.1.17 Select Standard Install
- 1.1.18 Geographic Region for Support: United States
- 1.1.19 Do not install 64-bit support
- 1.1.20 Choose Core System Support[1]
- 1.1.21 Do not preserve data, if applicable
- 1.1.22 No remote file systems
- 1.1.23 Customize the file system layout[2]

1	/	500 MB
2	swap	800 MB
3	/usr	2000 MB
4	/var	2000 MB
5	/opt	1000 MB
6	/export	1200 MB
7	/chroot	500 MB

- 1.1.24 Auto-reboot
- 1.1.25 Set root password[3]

[1] Choosing a minimal installation enhances security by excluding unneeded binaries and libraries.
[2] A few notes on file system sizes. Swap is 1.5 * RAM, /var is large for logging purposes, and /usr is large for source code and binaries. /export is where our group has standardized on for doing admin-type work: scripts, file transfers, tar backups, etc. /chroot is a separate file system for the jailed BIND binaries, logs, and config files.
[3] Something following the good password convention, i.e. an alpha-numeric/special character/CAPS password.

1.1.26 Networking configuration
    1.1.26.1 Create /etc/resolv.conf
    1.1.26.2 /bin/echo 'nameserver [IP address]' > /etc/resolv.conf
    1.1.26.3 /bin/chown root:root /etc/resolv.conf
    1.1.26.4 /bin/chmod 600 /etc/resolv.conf
1.1.27 Create /etc/defaultrouter
    1.1.27.1 /bin/echo '[gateway]' > /etc/defaultrouter
    1.1.27.2 /bin/chown root:root /etc/defaultrouter
    1.1.27.3 /bin/chmod 600 /etc/defaultrouter
1.1.28 Modify /etc/nsswitch.conf
    1.1.28.1 Make sure all entries are set to "files" except "hosts: files dns"
1.1.29 Reboot, plug in an ethernet cable, and try to ping the gateway.

Step 2 – Install any additional packages needed

Some additional packages will be needed going forward that are not included in the Core System Support installation. They are SUNWlibms (perl), SUNWntpr and SUNWntpu (ntp), SUNWadmc, SUNWadmfw, and SUNWlibC (showrev), and SUNWdoc (man pages). These packages can be found on disk 1 of the Solaris 8 media.

2.1 Mount CD[4]
    2.1.1 /bin/mkdir /mnt/cdrom
    2.1.2 /etc/mount –r –F hsfs /dev/dsk/c0t2d0s0 /mnt/cdrom
2.2 Find the packages on the media
    2.2.1 /bin/cd /mnt/cdrom/Solaris_8/Product
2.3 Install needed packages
    2.3.1 /usr/sbin/pkgadd –d . SUNWlibms SUNWntpr SUNWntpu SUNWadmc SUNWadmfw SUNWlibC SUNWdoc
2.4 Unmount CD
    2.4.1 /bin/cd /
    2.4.2 /etc/umount /mnt/cdrom

Step 3 – Patch the OS

Now that the server is on the internal test network, download the most recent recommended patch cluster for Solaris 8 (8_Recommended.zip[5] and the readme) from http://sunsolve.sun.com. While there, also download the patchdiag utility and the most recent patchdiag.xref file as well as patch-ID# 112438-01.

3.1 Read the patch cluster readme
3.2 Apply the patch cluster

[4] Manually mounting the cdrom when needed is a necessary hassle arising from doing a Core Support Installation. Vulnerabilities have been discovered in the past involving vold, so the daemon has been omitted from a core install.
[5] Download the file ftp://sunsolve.sun.com/pub/patches/CHECKSUMS. Execute md5sum 8_Recommended.zip and verify that it matches the entry in the CHECKSUMS file.

3.2.1 Copy the cluster to a suitable location
- 3.2.1.1 /bin/cp 8_Recommended.zip /var/tmp
- 3.2.1.2 /bin/cp 112438-01.zip /var/tmp

3.2.2 Take the machine to single-user mode to apply cluster
- 3.2.2.1 /sbin/init 0[6]
- 3.2.2.2 boot –s

3.2.3 Apply cluster patch
- 3.2.3.1 /bin/cd /var/tmp
- 3.2.3.2 /bin/unzip 8_Recommended.zip 112438-01.zip
- 3.2.3.3 /bin/cd 8_Recommended
- 3.2.3.4 ./install_cluster[7]
- 3.2.3.5 patchadd –d 112438-01[8]
- 3.2.3.6 reboot

3.2.4 Remove patch cluster
- 3.2.4.1 /bin/cd /var/tmp
- 3.2.4.2 /bin/rm –r 8_Recommended

3.2.5 Install patchdiag
- 3.2.5.1 /bin/cp patchdiag_1.0.4.tar.Z /export/sysadmin
- 3.2.5.2 /bin/cd /export/sysadmin
- 3.2.5.3 /bin/zcat patchdiag_1.0.4.tar.Z | /bin/tar xvf –
- 3.2.5.4 /bin/cp patchdiag.xref /export/sysadmin/patchdiag-1.0.4
- 3.2.5.5 /bin/cd patchdiag-1.0.4
- 3.2.5.6 ./patchdiag_setup

3.2.6 Run patchdiag and capture the report
- 3.2.6.1 /usr/bin/pkginfo > pkginfo.out
- 3.2.6.2 /usr/bin/showrev > showrev.out
- 3.2.6.3 ./patchdiag –p pkginfo.out showrev.out 5.8 sparc > patches_needed.out
- 3.2.6.4 analyze patches_needed.out[9]

3.2.7 Download any additional patches from http://sunsolve.sun.com
- 3.2.7.1 /bin/cp patches/*.zip /var/tmp/patches

---

[6] This is a step that should always be performed when applying a patch cluster. Going from run-level 6 to single-user mode can sometimes leave certain processes running or certain files open. This extra step ensures that this will not be the case.

[7] Certain patches will not install because "One or more patch packages are not installed on this system." This is normal, but an exact account of which patches failed to install can be found in /var/sadm/install_data/Solaris_8_recommended_log.

[8] Not included in the recommended patch cluster is a kernel patch which includes a random number generator /dev/random. OpenSSH depends on good, unpredictable numbers for generating keys, performing digital signatures and forming cryptographic challenges. If the random numbers that it uses are predictable, then the strength of the whole system is compromised. In this case, I prefer to use the "built-in" random number support for Solaris rather than fall back to a shareware solution that involves another installation, like PRNGd or EGD.

[9] Some would argue that this step is overkill after installing the patch cluster, but I like to be sure about the current patch level of a box when I build it. Patchdiag will tell you what failed during the install of the patch cluster, much like the log output in /var/sadm/install_data/Solaris_8_recommended_log. However, the format of the patchdiag report is much more readable and useful in this task because of the synopsis field provided in the report. It allows you to quickly evaluate whether any of these uninstalled patches are truly needed without having to download each and every patch readme. An example of a representative patchdiag report can be found in Appendix A.

3.2.8 Take the machine to single-user mode to apply patches
- 3.2.8.1 /sbin/init 0
- 3.2.8.2 boot –s

3.2.9 Apply all patches with a quick for loop
- 3.2.9.1 /bin/cd /var/tmp/patches
- 3.2.9.2 for i in `ls *.zip`
- 3.2.9.3 do
  unzip $i;
  patchadd $i >> output;
  done
- 3.2.9.4 analyze the output file[10]
- 3.2.9.5 /bin/cd ..
- 3.2.9.6 /bin/rm –r patches
- 3.2.9.7 reboot

Note on patching

The importance of properly patching a system is something that may get overlooked from time to time. In the case of Solaris, Sun conveniently provides a recommended patch cluster that will address most system stability and security issues that might ever arise on a system as a result of the patch level. So diligently patching a Solaris system is somewhat less tedious a chore than other systems, like Windows 2000.

Recently there have been a slew of vulnerabilities discovered that affect the Solaris operating system. Vulnerabilities like CDE ToolTalk and all the RPC related holes, the XDR library, and the DNS resolver library are just a few that have been recently addressed by Sun with patches. Often these special patch releases do not appear in the most recent patch clusters right away, so keep up with CERT and patch!

Note on compiling

When securing a Solaris server, the decision to perform a Core System Support installation of the OS has a far-reaching impact. This stripped-down version of the OS is by its very nature more streamlined and secure. However, in the interests of security, there are a few things you'll have to learn to live without on this version of the OS that deserve mention.

On of the first things one notices is that most of the standard Sun libraries and include files that are present on most machines, have been omitted here. This makes compiling applications from source, a very necessary step in further securing the server, an issue. The solution is simple: build another box in the test network that will serve as the build server for this machine. To do that, simply repeat steps 1, 2, and 3 above on a separate machine, changing step 1.1.20 to Entire Distribution plus OEM Support. There are no hardware limitations to consider for this build machine except to note that it should be of the same architecture as the hardened server, i.e. a sparc Ultra 5.

[10] Again, certain patches may not install because they either don't apply to the hardware system or the software they apply to is not installed on the system. This is normal.

Step 4 – Harden the OS

"Hardening the OS" is a catchall term for making various types of configuration changes to a wide range of services, settings, and applications. There are quite a few steps in this stage, and to simplify things (not to mention reducing the element of human error somewhat) I've automated the more straightforward and tedious steps by scripting them. Footnotes are used heavily and as a convenience in this step-by-step outline, and they detail the reasoning behind some of the steps. Appendices are also provided at the end of this paper for script and configuration examples.

- 4.1 Copy sysadmin scripts to /export/sysadmin/
- 4.2 Disable unneeded services at boot[11]
  - 4.2.1 /export/sysadmin/disable_services_at_boot.sh[12]
- 4.3 Create a startup script to set the umask[13]
  - 4.3.1 /bin/touch /etc/init.d/umask
  - 4.3.2 /bin/echo "#!/sbin/sh" >> /etc/init.d/umask
  - 4.3.3 /bin/echo "umask 022" >> /etc/init.d/umask
  - 4.3.4 /bin/chown root:sys /etc/init.d/umask
  - 4.3.5 /bin/chmod 744 /etc/init.d/umask
  - 4.3.6 /bin/ln –s /etc/init.d/umask /etc/rc2.d/S00umask
- 4.4 Hardening the TCP/IP stack
  - 4.4.1 /bin/mv /export/sysadmin/nddconfig[14] /etc/init.d
  - 4.4.2 /bin/chmod 744 /etc/init.d/nddconfig
  - 4.4.3 /bin/chown root:sys /etc/init.d/nddconfig
  - 4.4.4 /bin/ln –s /etc/init.d/nddconfig /etc/rc2.d/S70nddconfig
  - 4.4.5 Set TCP_STRONG_ISS=2 in /etc/default/inetinit[15]
  - 4.4.6 Protect against stack-smash attacks[16]
    - 4.4.6.1 /bin/echo "set noexec_user_stack = 1" >> /etc/system
    - 4.4.6.2 /bin/echo "set noexec_user_stack_log = 1" >> /etc/system
- 4.5 Access controls
  - 4.5.1 Modify values in /etc/default/login
    - 4.5.1.1 Make sure the line CONSOLE=/dev/console[17] is uncommented

[11] The best defense against a vulnerability that may affect a certain service is to turn that service off. It will be challenging enough securing the services that must be enabled. Identifying all *unneeded* services and disabling them is a step that will undoubtedly save you time and trouble down the road.

[12] This script does exactly what one might think: it disables unneeded services at boot by renaming files in /etc/rc*. An example of this script can be found in Appendix B.

[13] This step ensures that the startup scripts will run with the proper umask and that permissions on files that are subsequently created on a system are more restrictive.

[14] This script makes several changes to network settings that enhance security by setting options that will foil many known DDoS exploits. An example of this script (with security comments) can be found in Appendix C.

[15] By turning on strong initial sequence support you significantly reduce the likelihood that an attacker will be able to predict TCP/IP initial sequence numbers, or thereby effectively execute sequence number-based attacks.

[16] Enabling hardware protection for buffer overflow exploits will prevent an attacker from running shellcode in the stack. However, the heap is still executable, as are other areas of memory. The stack is still vulnerable, but it helps.

[17] Disabling root access to the system, even at the console, is a good practice because it forces use of local user accounts and thereby enhances auditing via logs.

4.5.1.2 Uncomment and set UMASK=022
4.5.2 Create /etc/ftpusers file[18]
4.5.3 cat /etc/passwd | cut –d ':' –f1 > /etc/ftpusers
4.5.4 chown root:sys /etc/ftpusers
4.5.5 chmod 600 /etc/ftpusers
4.6 The inetd daemon[19]
4.6.1 Modify /etc/init.d/inetsvc and /etc/rc2.d/S72inetsvc
4.6.1.1 Comment out line saying /usr/sbin/inetd –s &
4.6.1.2 Then modify line to #/usr/sbin/inetd –s –t &[20]
4.6.2 Comment out services in /etc/inetd.conf file[21]
4.7 Account administration
4.7.1 Disable unnecessary accounts[22]
4.7.1.1 Ensure that the password field of the /etc/shadow file says "NP" for these accounts: daemon, bin, sys, adm, lp, uucp, nuucp, listen, nobody, noaccess, and nobody4.
4.7.2 Make the shell on all non-root accounts /dev/null[23]
4.7.2.1 Edit the shell field of the /etc/passwd file to say /dev/null
4.7.3 Create wheel group[24]
4.7.3.1 /usr/sbin/groupadd -g 666 wheel
4.7.3.2 /bin/chgrp wheel /usr/bin/su /sbin/su.static
4.7.3.3 /bin/chmod 4550 /usr/bin/su /sbin/su.static
4.7.4 Create necessary user accounts and groups
4.7.4.1 Create admin accounts[25]
4.7.4.1.1 /usr/sbin/useradd –g 14 –G 666 –u 500 –d /export/sysadmin –c 'John Worthing' jworthin
4.7.4.2 Create smmsp account for sendmail[26]
4.7.4.2.1 /usr/sbin/groupadd –g 25 smmsp[27]
4.7.4.2.2 /usr/sbin/useradd –g 25 –u 1000 –c 'sendmail' –d /dev/null –s /bin/false smmsp

[18] This is just a precaution since all file transfer for this system will be accomplished via sftp/scp. In Solaris 8, this step has already been performed for you, but the file only contains the default users listed in /etc/passwd. It will be necessary to revisit this file after accounts have been added, entering at least the smmsp, sshd, and named accounts.
[19] Since there will be no internet standard services running on this system, inetd will be disabled. However, this housekeeping step ensures that in the event inetd is ever invoked, it will run a securely as possible.
[20] The –t option for inetd enables logging for all TCP services. The –s option runs inetd in normal standalone mode.
[21] With an editor, go through this file and comment out every single line so that no services are run by inetd.
[22] In Solaris 8, all of these legacy accounts have had this step performed for you already, out of the box. Instead of NP, the password field of /etc/shadow says *LK*, which accomplishes the same thing.
[23] This prevents anyone gaining access via these accounts from getting a shell to work within.
[24] Execution of the su command can be controlled by adding and configuring a wheel group. Only users who are members of the wheel group can execute su.
[25] Several user accounts for the various admins in our group need to be created, but I've just shown myself here. The notable configuration is that I'm adding my account to the primary group sysadmin and secondary group wheel.
[26] Installation and configuration of sendmail will be covered in-depth in a later section of the paper, but this is a step that we can touch on now. Sendmail can be compiled and configured into a binary that is not setuid root. smmsp is a pseudo-account that is used to accomplish this and should not be used by other daemons, and must be locked as well as contain an invalid default shell (/bin/false).
[27] You may get a warning for using group ID 25. Just ignore it, group ID 25 is recommended by the sendmail developers as stated in the readme of the 8.12.5 version of sendmail.

- 4.7.4.3 Create sshd user for privilege separation in OpenSSH[28]
  - 4.7.4.3.1 /bin/mkdir /var/empty
  - 4.7.4.3.2 /bin/chown root:sys /var/empty
  - 4.7.4.3.3 /bin/chmod 755 /var/empty
  - 4.7.4.3.4 /usr/sbin/groupadd –g 26 sshd
  - 4.7.4.3.5 /usr/sbin/useradd –g 26 –u 1001 –c 'sshd privsep' –d /var/empty –s /bin/false sshd
- 4.7.5 Password aging[29]
  - 4.7.5.1 Set MAXWEEKS=13 in /etc/default/passwd
- 4.8 File system configurations[30]
  - 4.8.1 Modify the mount options column in /etc/vfstab
    - 4.8.1.1 Make /usr ro[31]
    - 4.8.1.2 Make /var nosuid
    - 4.8.1.3 Make /opt nosuid,ro
- 4.9 Additional Logging
  - 4.9.1 Syslogd and centralized logging[32]
    - 4.9.1.1 Maximize logging in /etc/syslog.conf[33]
      - 4.9.1.1.1 /bin/echo “mail.debug\t\t\t/var/log/syslog” >> /etc/syslog.conf
      - 4.9.1.1.2 /bin/echo “*.info;mail.none\t\t\t/var/adm/messages” >> /etc/syslog.conf
    - 4.9.1.2 Send logs to the log server[34]
      - 4.9.1.2.1 /bin/echo “local2.debug\t\t\t@logserver” >> /etc/syslog.conf
      - 4.9.1.2.2 /bin/echo “*.info;kern.none;mail.none;local2.none\t\t\t @logserver” >> /etc/syslog.conf
      - 4.9.1.2.3 /bin/echo “mail.debug\t\t\t\t@logserver” >> /etc/syslog.conf
      - 4.9.1.2.4 /bin/echo “*.notice;kern.debug;*.err\t\t\t@logserver” >> /etc/syslog.conf

---

[28] Installation and configuration of OpenSSH will be covered in-depth in a later section of the paper, but this is a step that we can touch on now. When privsep is enabled, during the pre-authentication phase sshd will chroot to /var/empty and change its privileges to the sshd user. sshd is a pseudo-account that should not be used by other daemons, and must be locked as well as contain an invalid default shell (/bin/false).

[29] Changing passwords on a regular basis is always a good idea, but for some companies (like mine), it’s policy. Every three months users are required to change their passwords, even the administrators!

[30] There are two main levels of protection we can enforce when file systems mount at boot time. First, you want to guard against trojans and root kits infiltrating /usr, /opt, and /chroot, so they should be mounted read-only. Second, setuid scripts should be prevented from executing on these filesystems wherever possible.

[31] After the next reboot several key filesystems will be mounted as read-only filesystems. Any subsequent work within those filesystems can be accomplished without a reboot via the remount ufs option of the mount command. After the work is complete however, the filesystem cannot be returned to a read-only state without a reboot.

[32] Centralized logging is beneficial because it allows you to log everything in two places, locally and on a remote log server. This makes it very difficult for an intruder to hide his or her activities from the logs. It also has the added benefit of centralizing swatch log monitoring on the log server. Swatch is a perl script that monitors logs for certain events and then executes actions based on those events. By dumping all logs from all hosts to a log server, swatch only has to be installed and maintained on the log server. Big bonus for administrators!

[33] This will log mail entries to /var/log/syslog and everything else to /var/adm/messages, which is the default.

[34] This sends sudo logs as well as most kernel and mail messages to a logserver. Installation and configuration of sudo will be covered in-depth in a later section of the paper, but for now we can touch on it by saying it can be configured to log both locally and remotely using syslog. An example of syslog.conf can be found in Appendix D.

4.9.1.3 Update /etc/hosts file
4.9.1.3.1 /bin/echo "[logserver_IP] logserver" >> /etc/hosts
4.9.1.4 Update swatch configuration on logserver[35]
4.9.2 Configure syslogd to *not* listen on udp port 514 with the –t flag[36]
4.9.2.1 syslogd –t >/dev/msglog 2>&1 & in /etc/init.d/syslog
4.9.2.2 syslogd –t >/dev/msglog 2>&1 & in /etc/rc2.d/S72syslog
4.9.3 Log failed login attempts
4.9.3.1 /bin/touch /var/adm/loginlog
4.9.3.2 /bin/chown root:sys /var/adm/loginlog
4.9.4 Set permissions on logs
4.9.4.1 /bin/chmod 600 /var/adm/messages /var/log/syslog /var/adm/loginlog
4.9.5 Rotate logs daily and archive last 30 days
4.9.5.1 Update the newsyslog[37] script and modify root's cron
4.9.5.1.1 /bin/echo "10 3 * * * /usr/lib/newsyslog" >> /var/spool/cron/crontabs/root
4.10 Permissions on files and directories
4.10.1 Download the fix-modes script source by Casper Dik[38]
4.10.1.1 http://www.sun.com/solutions/blueprints/tools/FixModes.html
4.10.2 Install GCC on the build machine[39]
4.10.2.1 /bin/gunzip gcc-2.95.3-sol8-sparc-local.gz
4.10.2.2 /usr/sbin/pkgadd –d gcc-2.95.3-sol8-sparc-local
4.10.3 Build the fix-modes source on the build machine and tar it up
4.10.3.1 /bin/zcat FixModes.tar.Z | /bin/tar xvf –
4.10.3.2 /bin/cd FixModes
4.10.3.3 /usr/ccs/bin/make CC=gcc
4.10.3.4 /bin/tar cvf FixModes.tar ./fix-modes ./pmodes ./secure-modes
4.10.4 Untar fix-modes on hardened machine and run script
4.10.4.1 /bin/cd /export/sysadmin
4.10.4.2 /bin/tar xvf FixModes.tar
4.10.4.3 /bin/cd FixModes
4.10.4.4 ./fix-modes[40]
4.11 Secure cron[41]
4.11.1 Create the files cron.allow and at.allow

[35] Logs from all servers are dumped into three files on the log server: /var/log/sudo.log, /var/log/syslog, and /var/adm/messages. These three files are separately monitored by Swatch, and the Swatch configuration file may need updating, depending on the role of the remote server. An example of .swatchrc can be found in Appendix E.
[36] This disables syslogd's ability to receive connections, but not to initiate streams to the logserver, or log locally.
[37] An example of the modified newsyslog script can be found in Appendix F.
[38] This tool consists of a set of scripts which modify file permissions in an effort to make things more secure. The tool removes group/world write permissions on files, devices, and directories listed in the /var/sadm/install/contents, with the exception of those files listed in exceptions.h, and changes ownership of most files to root.
[39] This step is included here for completeness, but going forward it will be considered a step that has already been performed where steps that involve compiling source on the build machine are concerned. The gcc package for sparc Solaris 8 can be found at http://www.sunfreeware.com/.
[40] The fix-modes script can be run with various options that either make the changes to permissions more stringent or more flexible. The script can also be run in verbose mode, so redirect the output to a file for future analysis.
[41] The files in /etc/cron.d control which users can use the cron and at facilities.

4.11.1.1 /bin/echo "root" > /etc/cron.d/cron.allow
4.11.1.2 /bin/chown root:sys /etc/cron.d/cron.allow
4.11.1.3 /bin/chmod 600 /etc/cron.d/cron.allow
4.11.1.4 /bin/cp /etc/cron.d/cron.allow /etc/cron.d/at.allow
4.11.2 Recreate the files cron.deny and at.deny
4.11.2.1 /bin/cat /etc/passwd | /bin/cut -f1 -d: | /bin/grep -v root > /etc/cron.d/cron.deny
4.11.2.2 /bin/chown root:sys /etc/cron.d/cron.deny
4.11.2.3 /bin/chmod 600 /etc/cron.d/cron.deny
4.11.2.4 /bin/cp /etc/cron.d/cron.deny /etc/cron.d/at.deny
4.12 Time Synchronization[42]
4.12.1 Create /etc/inet/ntp.conf and add three public time servers
4.12.1.1 /bin/echo “server 204.152.184.72 # clock.isc.org” > /etc/inet/ntp.conf
4.12.1.2 /bin/echo “server 192.43.244.18 # time.nist.gov” > /etc/inet/ntp.conf
4.12.1.3 /bin/echo “server 128.9.176.30 # timekeeper.isi.edu” >> /etc/inet/ntp.conf
4.13 EEPROM security[43]
4.13.1 Turn EEPROM security on
4.13.1.1 /usr/sbin/eeprom security-mode=full[44]
4.13.2 Set EEPROM login attempts
4.13.2.1 /usr/sbin/eeprom security-#badlogins=3
4.13.3 Add EEPROM banner
4.13.3.1 /usr/sbin/eeprom oem-banner?=true
4.13.3.2 /usr/sbin/eeprom oem-banner=This is a restricted server. Unauthorized access is strictly prohibited.
4.14 Disclaimers[45]
4.14.1 cat > /etc/issue

```
********************** WARNING ****************************
                This is a restricted server.
Unauthorized access to this system is strictly prohibited.
***********************************************************
^C
```

4.14.2 cp /etc/motd /etc/motd.orig
4.14.3 cat /etc/issue /etc/motd.orig > /etc/motd

[42] The importance of accurate time keeping on a server was underscored heavily in a FIST course I took last year. In legal terms, the variance of a minute between the log entries on two different servers could be the difference between evidence and hearsay.
[43] Using stop-a, or break, anyone can interrupt your machine and reboot it from cdrom (or their disk), and have complete access to your files. Physical security has a lot to do with the success of such an attack, but enabling password protection on the EEPROM can help stop this kind of attack.
[44] One thing to remember here is not to forget the password. You will not be able to change it unless you have access to the running system. If you find yourself locked out of the EEPROM and unable to reboot the system, you may have to replace some hardware.
[45] Again, as a matter of legality, servers that are restricted to authorized users only should display this fact as prominently as possible. Otherwise, an unauthorized user gaining access to your system may have an argument in his or her defense in a court of law.

Step 5 – Compile, install, and configure sudo

sudo is a utility many administrators use to accomplish the task of giving certain users elevated privileges on a system without allowing the root password to be known by many, which is never a good idea from the view point of security. It also has the added feature of an audit trail. Centralized logging is accomplished in sudo via syslogd and can greatly simplify the task of troubleshooting problems on a server with so many users having elevated privileges.

5.1 Download sudo 1.6.6 source code from http://www.sunfreeware.com/
5.2 Copy the source code to the build machine and untar
  5.2.1 /bin/cp sudo-1.6.6.tar.gz /export/sysadmin
  5.2.2 /bin/gunzip –dc sudo-1.6.6.tar.gz | /bin/tar xvf –
5.3 Read the INSTALL file for information on options for compiling sudo
  5.3.1 /bin/cd sudo-1.6.6
  5.3.2 /bin/vi INSTALL
5.4 Run ./configure with all the options needed to build sudo properly
  5.4.1 ./configure \
    --with-mailto=jworthing@jii.com,admin2@jii.com,etc. \
    --with-mail-if-noperms \
    --with-mailsubjects="*** SUDO user alert on %h ***" \
    --with-logging=both[46] \
    --with-logpath=/var/log/sudo.log \
    --with-all-insults \
5.5 Compile sudo and install the binaries
  5.5.1 /usr/ccs/bin/make
  5.5.2 /usr/ccs/bin/make check
  5.5.3 /usr/ccs/bin/make install
  5.5.4 /usr/ccs/bin/make clean
  5.5.5 /usr/ccs/bin/make distclean
5.6 Configure sudoers file
  5.6.1 /usr/local/sbin/visudo[47]
5.7 Tar up binaries and configs and move them to hardened machine
  5.7.1 /bin/tar cvf ./sudo_bin.tar /usr/local/bin/sudo /usr/local/sbin/visudo /etc/sudoers
5.8 Untar and double-check permissions
  5.8.1 /bin/tar xvf sudo_bin.tar
  5.8.2 /bin/chmod 4111 /usr/local/bin/sudo
  5.8.3 /bin/chmod 111 /usr/local/sbin/visudo
  5.8.4 /bin/chmod 440 /etc/sudoers

[46] This is where you set sudo to log in multiple places. Logging sudo to /var/log/sudo.log is a fairly common practice, but this setting also allows sudo to log to syslogd as well. In step 4.9.1.2.1 syslogd was configured to send sudo logs to the logserver as well, so this piece is already configured.

[47] visudo is sudo's tool for editing the sudo configuration file /etc/sudoers. The format of this file and the options available to sudo are discussed at length in the man pages that get installed. On this server, sudo's main objectives are to protect the root password and provide an audit trail of those who use it. So the configuration is fairly generic with an Admin group containing all admin accounts. An example of /etc/sudoers can be found in Appendix G.

Step 6 – Compile, install, and configure sendmail

sendmail is required on this server because the machine needs to be able to send administrators e-mails, for a variety of reasons. It is a good idea to replace Sun's sendmail binary with one built from source, so here we go.

6.1 Download sendmail 8.12.6 source code from http://www.sendmail.org/
6.2 Copy the source code to the build machine and untar
  6.2.1 /bin/cp sendmail.8.12.6.tar.gz /export/sysadmin
  6.2.2 /bin/gunzip –dc sendmail.8.12.6.tar.gz | /bin/tar xvf –
6.3 Read the INSTALL file for information on compiling sendmail
  6.3.1 /bin/cd sendmail-8.12.6
  6.3.2 /bin/vi INSTALL
6.4 Build the sendmail binaries
  6.4.1 /bin/cd sendmail
  6.4.2 sh Build
6.5 Build the sendmail configuration files and install them in /etc/mail[48]
  6.5.1 /bin/cd ../cf/cf
  6.5.2 /bin/cp generic-solaris.mc sendmail.mc
  6.5.3 sh Build sendmail.cf
  6.5.4 sh Build install-cf
6.6 Install the sendmail binaries[49]
  6.6.1 /bin/cd ../../sendmail
  6.6.2 sh Build install
6.7 Tar up binaries and configs and move them to hardened machine
  6.7.1 /bin/tar cvf ./sendmail_bin.tar /usr/lib/sendmail /etc/mail
6.8 Untar and double-check permissions on sendmail binary
  6.8.1 /bin/tar xvf sendmail_bin.tar
  6.8.2 /bin/chown root:smmsp /usr/lib/sendmail
  6.8.3 /bin/chmod 2555 /usr/lib/sendmail
6.9 Create the world-write able mail folder and tweak permissions
  6.9.1 /bin/mkdir /var/spool/clientmqueue
  6.9.2 /bin/chown smmsp:smmsp /var/spool/clientmqueue
  6.9.3 /bin/chmod 770 /var/spool/clientmqueue
  6.9.4 /bin/chown root:wheel /var/spool/mqueue
  6.9.5 /bin/chmod 700 /var/spool/mqueue
  6.9.6 /bin/chown root:wheel /etc/mail/sendmail.cf /etc/mail/submit.cf
  6.9.7 /bin/chmod 444 /etc/mail/sendmail.cf /etc/mail/submit.cf
6.10 Edit sendmail startup script so that it doesn't listen on port 25
  6.10.1 Change MODE="-bd" to MODE="" in /etc/rc2.d/S88sendmail
  6.10.2 Change all instances of QUEUEINTERVAL="15m" to "1m"

[48] Since this server will neither receive mail nor relay it, the configuration of sendmail is greatly simplified. There is no need to for a relay-domains file or a local-host-names file because of this. The default sendmail.cf that gets created is adequate for the task of sending mail only. Make any changes necessary in sendmail.cf to ensure mail delivery. Create any mail aliases you feel are useful in /etc/mail/aliases and issue the command /bin/newaliases.
[49] There are other binaries that come in the sendmail source that can be built and installed (i.e., makemap, mailstats, etc.), but this is not necessary. The Sun versions of these binaries will work fine.

Step 7 – Compile, install, and configure OpenSSH

OpenSSH is fast becoming the de facto industry standard for remote shell administration in UNIX. It is commonly used as a replacement for telnet and ftp because it encrypts the connection between hosts, eliminating the age-old security concern associated with transmitting the password in the clear. There are a slew of other good reasons to use OpenSSH (like its secure-cp and secure-ftp file transfer mechanisms and its secure-rsh feature), but 3DES encryption over the wire is reason enough.

There are several sub-steps involved when compiling OpenSSH from source that involve compiling some other utilities first. All compiling steps will be covered here.

7.1 Download the following code distributions from http://www.sunfreeware.com/
- wget 1.8.2: wget-1.8.2-sol8-sparc-local.gz
- tcp wrappers 7.6: tcp_wrappers-7.6.tar.gz
- zlib 1.1.4: zlib-1.1.4.tar.gz
- openssl 0.9.6g: openssl-0.9.6g.tar.gz
- openssh 3.4p1: openssh-3.4p1.tar.gz

7.2 Copy the source code to the build machine
- 7.2.1 /bin/cp wget-1.8.2-sol8-sparc-local.gz tcp_wrappers-7.6.tar.gz zlib-1.1.4.tar.gz openssl-0.9.6g.tar.gz openssh-3.4p1.tar.gz /export/sysadmin

7.3 Unzip wget and install the package
- 7.3.1 /bin/cd /export/sysadmin
- 7.3.2 /bin/gunzip wget-1.8.2-sol8-sparc-local.gz
- 7.3.3 /bin/pkgadd –d ./wget-1.8.2-sol8-sparc-local

7.4 Untar tcp wrappers and tweak the Makefile
- 7.4.1 /bin/gunzip –dc tcp_wrappers-7.6.tar.gz | /bin/tar xvf –
- 7.4.2 /bin/cd tcp_wrappers_7.6
- 7.4.3 /bin/vi Makefile
  - 7.4.3.1 Under the advanced installation section, for Solaris 2.x, uncomment REAL_DAEMON_DIR=/usr/sbin
  - 7.4.3.2 Under the sunos5 section, add CC=gcc to the 2nd line

7.5 Build the tcp wrappers binaries and libraries and copy needed files[50]
- 7.5.1 /usr/ccs/bin/make
- 7.5.2 /bin/cp tcpd safe_finger /usr/sbin
- 7.5.3 /bin/cp libwrap.a /usr/lib
- 7.5.4 /bin/cp tcpd.h /usr/include
- 7.5.5 /bin/cd ..

7.6 Untar zlib and read the README file
- 7.6.1 /bin/gunzip –dc zlib-1.1.4.tar.gz | /bin/tar xvf –
- 7.6.2 /bin/cd zlib-1.1.4
- 7.6.3 /bin/vi README

[50] For OpenSSH to be compiled with wrappers support, libwrap.a and tcpd.h have to be copied to a place where the OpenSSH installation's configure script can find them.

7.7 Run ./configure and then build zlib[51]
- 7.7.1 ./configure
- 7.7.2 /usr/ccs/bin/make test
- 7.7.3 /usr/ccs/bin/make install
- 7.7.4 /bin/cd ..

7.8 Untar OpenSSL and read the INSTALL file
- 7.8.1 /bin/gunzip –dc openssl-0.9.6g.tar.gz | /bin/tar xvf –
- 7.8.2 /bin/cd openssl-0.9.6g
- 7.8.3 /bin/vi INSTALL

7.9 Run ./config and then build OpenSSL[52]
- 7.9.1 ./config
- 7.9.2 /usr/ccs/bin/make
- 7.9.3 /usr/ccs/bin/make test
- 7.9.4 /usr/ccs/bin/make install
- 7.9.5 /bin/cd ..

7.10 Untar OpenSSH and read the INSTALL file
- 7.10.1 /bin/gunzip –dc openssh-3.4p1.tar.gz | /bin/tar xvf –
- 7.10.2 /bin/cd openssh-3.4p1
- 7.10.3 /bin/vi INSTALL

7.11 Run ./configure and then build OpenSSH
- 7.11.1 ./configure --with-tcp-wrappers
- 7.11.2 /usr/ccs/bin/make
- 7.11.3 /usr/ccs/bin/make install
- 7.11.4 /bin/cd ..

7.12 Tar up binaries and configs and move them to hardened machine
- 7.12.1 /bin/tar cvf ./openssh_bin.tar /usr/local/ssl /usr/local/libexec /usr/local/lib/libz.a /usr/local/sbin/sshd /usr/local/bin/ssh* /usr/local/bin/scp /usr/local/bin/sftp /usr/local/etc /usr/sbin/safe_finger /usr/sbin/tcpd

7.13 Untar and change ownership, but only verify permissions[53]
- 7.13.1 /bin/tar xvf openssh_bin.tar
- 7.13.2 /bin/chown –R root:sys /usr/local/ssl
- 7.13.3 /bin/chown –R root:sys /usr/local/libexec
- 7.13.4 /bin/chown –R root:sys /usr/local/etc
- 7.13.5 /bin/chown –R root:sys /usr/local/lib

[51] The zlib libraries are needed for the compression OpenSSH does.

[52] OpenSSL provides the mechanism with which OpenSSH encrypts its tunnel, thereby rendering the date therein unreadable to unauthorized folks. Encryption is a broad topic of discussion, so here it is sufficient to say that OpenSSL provides symmetric encryption, public key cryptography and key agreement, certificate handling, cryptographic hash functions and a cryptographic pseudo-random number generator. In addition, OpenSSL provides a means for improved authentication and access control at the connection level, reducing the chance of session hijacking or unauthorized access.

[53] Most of these binaries are 755, but not all of them. Listing all the permissions here would be prohibitive - suffice it to say that all permissions, directories and files alike, being untarred should be cross-referenced for accuracy against the originals on the build machine.

7.13.6 /bin/chown root:sys /usr/local/bin/ssh* /usr/local/bin/scp /usr/local/bin/sftp /usr/local/sbin/sshd /usr/sbin/safe_finger /usr/sbin/tcpd

7.14 Configure the ssh_config file

7.14.1 /bin/vi /usr/local/etc/ssh_config[54]

7.15 Configure the sshd_config file

7.15.1 /bin/vi /usr/local/etc/sshd_config[55]

7.16 Create /etc/hosts.allow file[56]

7.16.1 /bin/echo "sshd : [allowed_IPs_or_networks]" >> /etc/hosts.allow

7.16.2 /bin/chmod 600 /etc/hosts.allow

7.16.3 /bin/chown root:sys /etc/hosts.allow

7.17 Create /etc/hosts.deny file[57]

7.17.1 /bin/echo "ALL : ALL : (/usr/sbin/safe_finger –l @%h | /bin/mailx –s %d–%h jworthing@jii.com,admin2,etc.)" >> /etc/hosts.deny

7.17.2 /bin/chmod 600 /etc/hosts.deny

7.17.3 /bin/chown root:sys /etc/hosts.deny

7.18 Create a startup script for sshd

7.18.1 /bin/cp /export/sysadmin/sshd[58] /etc/init.d

7.18.2 /bin/chown root:sys /etc/init.d/sshd

7.18.3 /bin/chmod 744 /etc/init.d/sshd

7.18.4 /bin/ln –s /etc/init.d/sshd /etc/rc2.d/S86sshd

7.19 Start sshd and check that privilege separation is working

7.19.1 /etc/init.d/sshd start[59]

7.19.2 /bin/ps –ef | /bin/grep sshd[60]

```
root     264     1     0   Sep 05   ?  0:28   /usr/local/sbin/sshd
```

7.19.3 /usr/local/ssh ns2 (connect to the hardened server using ssh)

7.19.4 /bin/ps –ef | /bin/grep sshd[61]

```
root     264     1     0   Sep 05   ?  0:28   /usr/local/sbin/sshd
root     17233   264   0  19:16:28  ?  0:00   /usr/local/sbin/sshd
jworthin 17235   17233 0  19:16:31  ?  0:00   /usr/local/sbin/sshd
```

[54] An example of the ssh_config file, with security comments, can be found in Appendix H.

[55] An example of the sshd_config file, with security comments, can be found in Appendix I.

[56] The /etc/hosts.allow and /etc/hosts.deny files are a tcp wrappers convention, and a sound way of limiting internet access. Wrappers functionality is compiled into sshd, so it automatically refers to these files when a connection is requested. If other services (like those provided via inetd) are ever needed, wrappers can be used with them as well.

[57] This file denies access to all IPs and services not explicitly allowed in /etc/hosts.allow. A really useful tactic can be employed using safe_finger on connections that are denied by tcp wrappers. Denied connections can have safe_finger executed against them and that information is then passed on to the admins via e-mail. This is useful because you are immediately alerted when scans and other methods of prying are taking place.

[58] An example of a sshd startup script can be found in Appendix J.

[59] If privilege separation is not supported on the OS or doesn't have the proper sshd account created, sshd will error here. As an aside, this is also when you would see an error about PRNG not being seeded if patch 112438-01 for /dev/random has not been applied.

[60] Listing the sshd processes, you should see one sshd process started and owned by root.

[61] What you should see here are two more sshd processes for the connection that was just made, one owned by root and one owned by jworthin, all child processes of the original sshd process started by root. This is the chroot piece of privilege separation at work.

Step 8 – Compile, install, and configure BIND 9.2.2c1

BIND (Berkeley Internet Name Domain) is an implementation of DNS protocols and provides an open source reference implementation of the major components of the Domain Name System, including a DNS server, a DNS resolver library, and tools for verifying the proper operation of the DNS server.

BIND is used on the majority of name service machines on the Internet, providing a robust and stable architecture on top of which an organization's naming architecture can be built.

8.1 Download BIND version 9.2.2rc1 from http://www.isc.org/
8.2 Copy the source code to the build machine and untar
  8.2.1 /bin/cp bind-9.2.2rc1.tar.gz /export/sysadmin
  8.2.2 /bin/gunzip –dc bind-9.2.2rc1.tar.gz | /bin/tar xvf –
8.3 Read the README file for information on compiling BIND
  8.3.1 /bin/cd bind-9.2.2rc1
  8.3.2 /bin/vi README
8.4 Run ./configure with all the options needed to build BIND properly
  8.4.1 ./configure \
    --prefix=/opt/bind \
    --sysconfdir=/etc \
    --localstatedir=/var \
    --with-openssl[62] \
    --with-libtool \
    --enable-threads
8.5 Compile BIND and install the binaries
  8.5.1 /usr/ccs/bin/make
  8.5.2 /usr/ccs/bin/make test
  8.5.3 /usr/ccs/bin/make install
8.6 Tar up binaries and configs and move them to the hardened machine
  8.6.1 /bin/tar cvf ./bind_bin.tar /opt/bind
8.7 Untar and change ownership, but only verify permissions
  8.7.1 /bin/tar xvf bind_bin.tar
  8.7.2 /bin/chown –R root:sys /opt/bind
8.8 Configure BIND to run as root
  8.8.1 Create the BIND RunDirs
    8.8.1.1 /bin/mkdir -p /var/named/internal /var/named/external[63]
    8.8.1.2 /bin/chmod –R 1770 /var/named
    8.8.1.3 /bin/chown –R root:named /var/named
  8.8.2 Created a non-privileged bind user
    8.8.2.1 /usr/sbin/groupadd –g 27 named

[62] This option allows the use of SSL for DNSSEC and encrypted zone transfers.
[63] There are security comments in the named.conf file in Appendix K that speak about the significance of internal and external views.

8.8.2.2 /usr/sbin/useradd –g 27 –u 1002 –c 'named account' –d /var/named –s /bin/false named

8.8.3 Complete ownership

8.8.3.1 /bin/chown root:daemon /var/run

8.8.3.2 /bin/chmod 775 /var/run

8.8.4 Create /etc/rndc.conf[64]

8.8.4.1 /opt/bind/sbin/rndc-confgen > /etc/rndc.conf

8.8.4.2 /bin/chown root:sys /etc/rndc.conf

8.8.4.3 /bin/chmod 600 /etc/rndc.conf

8.8.5 Create /etc/rndc.key[65]

8.8.5.1 /bin/vi /etc/rndc.key

8.8.5.2 /bin/chown root:sys /etc/rndc.key

8.8.5.3 /bin/chmod 600 /etc/rndc.key

8.8.6 Create /etc/named.conf[66]

8.8.6.1 /bin/cp /export/sysadmin/named.conf /etc

8.8.6.2 /bin/chown root:sys /etc/named.conf

8.8.6.3 /bin/chmod 600 /etc/named.conf

8.8.7 Create the log files used by named as specified in /etc/named.conf

8.8.7.1 /bin/touch /var/adm/namedlog

8.8.7.2 /bin/touch /var/adm/namedstats

8.8.7.3 /bin/touch /var/adm/xferlog

8.8.7.4 /bin/chown root:sys /var/adm/namedlog /var/adm/namedstats /var/adm/xferlog

8.8.7.5 /bin/chmod 600 /var/adm/namedlog /var/adm/namedstats /var/adm/xferlog

8.8.8 Create localhost zone file[67]

8.8.8.1 /bin/cp /export/sysadmin/db.127.0.0 /var/named/internal

8.8.8.2 /bin/chmod 600 /var/named/internal/db.127.0.0

8.8.8.3 /bin/chown root:root /var/named/internal/db.127.0.0

8.8.9 Get root.hints file from the root servers

8.8.9.1 /bin/cd /var/named/

8.8.9.2 /usr/local/bin/wget ftp://internic.net/domain/named.root

8.8.9.3 /bin/mv named.root named.cache

8.8.9.4 /bin/chmod 600 named.cache

8.8.9.5 /bin/chown root:root named.cache

8.8.10 Add the commands above to cron to update named.cache file monthly

8.8.10.1 /bin/echo “0 0 1 * * [commands]” >>/var/spool/cron/crontabs/root

8.8.11 Edit /etc/resolv.conf and change 1st nameserver entry to 127.0.0.1

[64] BIND 9.x uses rndc, the successor to ndc, to control the name server. rndc uses an authenticated control channel to send messages to the name server. This is significant because it helps mitigate the risk of someone spoofing a message to the control channel.

[65] This key file is created simply by pasting the bottom portion of the /etc/rndc.conf file into it and removing the comments.

[66] Copy the named.conf from the DNS master server and make changes appropriate for a slave server. An example of named.conf, with security comments, can be found in Appendix K.

[67] Again, this file could be copied from the DNS master and changed for the slave. Without this file, step 8.8.11 will cause problems for this nameserver.

8.8.11.1 /bin/vi /etc/resolv.conf
8.8.12 Launch named and test
8.8.12.1 /opt/bind/sbin/named &
8.8.12.2 /opt/bind/bin/nslookup localhost
8.9 Configure BIND to run as the named user in a chroot jail[68]
8.9.1 Emulate BIND's directory structure under /chroot and copy files[69]
8.9.1.1 /bin/mkdir –p –m 755 /chroot/etc /chroot/var/run /chroot/var/adm
8.9.1.2 /bin/cp –r –p /var/named /chroot/var
8.9.1.3 /bin/cp –p /etc/named.conf /etc/rndc.conf /etc/rndc.key /chroot
8.9.1.4 /bin/cp –p /var/adm/logs/namedlog /var/adm/logs/namedstats /var/adm/logs/xferlog /chroot
8.9.1.5 /bin/mkdir /chroot/dev
8.9.1.6 /bin/cd /chroot/dev
8.9.1.7 /bin/mknod null c [major device number] [minor dev number][70]
8.9.1.8 /bin/chmod 666 null
8.9.1.9 /bin/mknod random c [major device number] [minor dev number]
8.9.1.10 /bin/cd /chroot
8.9.1.11 /bin/chown –R named:named *
8.9.2 Run named as non-privileged user in a chroot jail and check for errors
8.9.2.1 /opt/bind/sbin/named –u 1002 –t /chroot &
8.9.2.2 /bin/vi /chroot/var/adm/namedlog
8.9.3 Create a startup script for named
8.9.3.1 /bin/cp /export/sysadmin/named[71] /etc/init.d
8.9.3.2 /bin/chown root:sys /etc/init.d/named
8.9.3.3 /bin/chmod 744 /etc/init.d/named
8.9.3.4 /bin/ln –s /etc/init.d/named /etc/rc2.d/S50named
8.10 Test BIND in the chroot jail
8.10.1 Start named in a /chroot jail (if named is not started already)
8.10.1.1 /etc/init.d/named start
8.10.2 Check that file transfers happened with master DNS server
8.10.2.1 /bin/ls -la /chroot/var/named/internal /chroot/var/named/external
8.10.3 Check logs and query the local nameserver for an SOA record
8.10.3.1 /bin/tail /chroot/var/adm/namedlog
8.10.3.2 /opt/bind/sbin/nslookup www.jii.com

[68] Named can be run in a chrooted environment by specifying the –t option. This can help improve system security by placing BIND in a "jail", which will limit the damage done if the server is ever compromised. Another useful feature of BIND is the ability to run the daemon as a non-privileged user, and thereby limit the damage an unauthorized user can do if he or she ever gains a shell through named.

[69] In order for a chroot environment to work properly in a particular directory (for example, /chroot), you will need to set up an environment that includes everything BIND needs to run. From BIND's point of view, /chroot is the root of the filesystem. You will need to adjust the values of options like directory and pid-file to account for this. Unlike earlier versions of BIND, you will not need to compile named statically nor install shared libraries under the new root. However you will need to set up special files like /dev/null and /dev/random within the jail.

[70] Major and minor device numbers for these character special files can be obtained by doing an ls –l on the actual physical device files that the links /dev/null and /dev/random point to. On most systems, /dev/null has a major device number of "13," and a minor device number of "2". These device numbers often vary for /dev/random on different systems, so always check with ls before creating a character special file.

[71] An example of this startup script can be found in Appendix L.

Step 9 – Compile, install, and configure Nessus

Nessus is a free, powerful, and easy to use security scanner that can be used to remotely audit a given network, or single host or hosts, to determine its various vulnerabilities to hacker attacks. Other than the obvious fact that Nessus is free, another benefit of this tool is that it makes no assumptions about what services are running on what ports. Nessus has a nice gui and can create reports in a wide variety of formats.

- 9.1 Download the four parts of Nessus version 1.2.5 from http://www.nessus.org/
- 9.2 Copy the source code to the build machine
  - 9.2.1 /bin/cp nessus-libraries.tar.gz libnasl-1.2.5.tar.gz nessus-core-1.2.5.tar.gz nessus-plugins-1.2.5.tar.gz /export/sysadmin
- 9.3 Read the Installation Instructions on the web site for info on compiling Nessus
  - 9.3.1 Untar and compile the Nessus libraries
    - 9.3.1.1 /bin/gunzip –dc nessus-libraries-1.2.5.tar.gz | /bin/tar xvf –
    - 9.3.1.2 /bin/cd nessus-libraries
    - 9.3.1.3 ./configure && /usr/ccs/bin/make && /usr/ccs/bin/make install
    - 9.3.1.4 /bin/cd ..
  - 9.3.2 Untar and compile the libnasl libraries
    - 9.3.2.1 /bin/gunzip –dc libnasl-1.2.5.tar.gz | /bin/tar xvf –
    - 9.3.2.2 /bin/cd libnasl
    - 9.3.2.3 ./configure && /usr/ccs/bin/make && /usr/ccs/bin/make install
    - 9.3.2.4 /bin/cd ..
  - 9.3.3 Untar and compile the Nessus core
    - 9.3.3.1 /bin/gunzip –dc nessus-core-1.2.5.tar.gz | /bin/tar xvf –
    - 9.3.3.2 /bin/cd nessus-core
    - 9.3.3.3 ./configure && /usr/ccs/bin/make && /usr/ccs/bin/make install
    - 9.3.3.4 /bin/cd ..
  - 9.3.4 Untar and compile the Nessus plug-ins
    - 9.3.4.1 /bin/gunzip –dc nessus-plugins-1.2.5.tar.gz | /bin/tar xvf –
    - 9.3.4.2 /bin/cd nessus-plugins
    - 9.3.4.3 ./configure && /usr/ccs/bin/make && /usr/ccs/bin/make install
    - 9.3.4.4 /bin/cd ..
- 9.4 Tar up binaries and configs and move them to the hardened machine
  - 9.4.1 /bin/tar cvf ./nessus_bin.tar /usr/local/bin/nessus* /usr/local/sbin/nessus* /usr/local/lib/lib* /usr/local/lib/nessus /usr/local/man /usr/local/etc/nessus
- 9.5 Untar and change ownership, but only verify permissions
  - 9.5.1 /bin/tar xvf nessus_bin.tar
  - 9.5.2 /bin/chown –R root:sys /usr/local/bin /usr/local/sbin /usr/local/lib /usr/local/man /usr/local/etc/nessus
- 9.6 Configure the Nessus daemon, if needed
  - 9.6.1 /bin/vi /usr/local/etc/nessus/nessusd.conf[72]

[72] Several options for nessusd can be configured in this file, but the standard configuration file will suffice here.

Step 10 – Compile, install, and configure Tripwire ASR

Tripwire is a system integrity analyzer first developed in 1992. Tripwire automatically monitors changes to files and system attributes, including file size, access flags, write time, etc. Since its shareware days it has gone widely commercial and can only be obtained for free if you wish to employ the older academic source release (ASR). For the needs of this server, this is more than adequate.

10.1 Download tripwire ASR from http://www.tripwire.com/downloads
10.2 Copy the source code to the build machine and untar
  10.2.1 /bin/cp Tripwire-1.3.1-2.tar.gz /export/sysadmin
  10.2.2 /bin/gunzip –dc Tripwire-1.3.1-2.tar.gz | /bin/tar xvf –
10.3 Read the README file for information on compiling tripwire
  10.3.1 /bin/cd Tripwire-1.3.1-2
  10.3.2 /bin/vi README
10.4 Modify the config.h file so that tripwire installs on /opt[73]
  10.4.1 /bin/vi include/config.h[74]
10.5 Modify the Makefile so that tripwire will build and install correctly
  10.5.1 /bin/vi Makefile[75]
10.6 Compile tripwire 1.3.1 ASR and install the binaries and database
  10.6.1 /usr/ccs/bin/make all
  10.6.2 /usr/ccs/bin/make test
  10.6.3 /usr/ccs/bin/make install
10.7 Tar up binaries and configs and move them to the hardened machine
  10.7.1 /bin/tar cvf ./tripwire_bin.tar /opt/tw
10.8 Untar and change ownership, but only verify permissions
  10.8.1 /bin/tar xvf ./tripwire_bin.tar
  10.8.2 /bin/chown –R root:sys /opt/tw
10.9 Create a suitable tw.config file for tripwire
  10.9.1 /bin/vi /opt/tw/bin/tw/tw.config[76]
  10.9.2 /bin/chown root:sys /opt/tw/bin/tw/tw.config
  10.9.3 /bin/chmod 600 /opt/tw/bin/tw/tw.config
10.10 Add a daily tripwire scan to cron
  10.10.1 /bin/cp /export/sysadmin/tw.check /opt/tw/tw.check[77]
  10.10.2 /bin/chmod 700 /opt/tw/tw.check
  10.10.3 /bin/chown root:sys /opt/tw/tw.check

[73] We do this because /opt will be mounted as a read-only filesystem, and this helps ensure file integrity. This is a perfect place for the tripwire database to reside, because data from tripwire is only as trustworthy as its database.
[74] Settings in this file govern where the tripwire binaries and database will eventually reside. An example of config.h can be found in Appendix M.
[75] Settings in this file govern where the tripwire binaries and database will reside, as well as other Solaris-specific settings that allow tripwire to compile properly. An example of the Makefile can be found in Appendix N.
[76] This is the tripwire configuration file that tells tripwire what files to monitor and how to build its database. An example of tw.config, with security comments, can be found in Appendix O.
[77] This script automates the task of daily integrity checks of the hard disk using tripwire. An example of tw.check can be found in Appendix P.

10.10.4 /bin/echo "0 1 * * * /opt/tw/tw.check">> /var/spool/cron/crontabs/root
10.11 Create the tripwire database
10.11.1 Take the server to single-user mode
10.11.1.1 /sbin/init 0
10.11.1.2 boot –s
10.11.2 Mount the remaining file systems
10.11.2.1 /etc/mount –F ufs –o rw /dev/dsk/c0t0d0s5 /opt
10.11.2.2 /etc/mount –F ufs /dev/dsk/c0t0d0s6 /export
10.11.2.3 /etc/mount –F ufs /dev/dsk/c0t0d0s7 /chroot
10.11.3 Initialize the database and copy it in place
10.11.3.1 /opt/tw/bin/tripwire --initialize[78]
10.11.3.2 /bin/cp /opt/tw/bin/databases/tw.db_ns2 /opt/tw/var/tripwire
10.11.4 Make a hardcopy of the database[79]
10.12 Reboot

- **Ongoing Maintenance**

Once the step-by-step checklist was completed, the server was re-IPed and reconfigured with the network settings required by the co-location facility. The box was then shipped to the co-location facility and installed. It came up quickly and started answering DNS queries right away. Success!

So the next step is to outline a strategy for ongoing maintenance that will ensure the continued smooth operation of this server at the co-location facility. What follows is an outline of the procedures to be periodically followed, broken down by the frequency in which they should be performed.

Network, Log, and Process Monitoring -- Constantly

- Several SiteScope alerts will be configured on our network monitoring server here to test the availability of certain network services on that machine there. The first and most obvious test is a ping, simply to determine whether the server is reachable. The second is a DNS test – several DNS tests actually. Since this slave server only serves a handful of domains, each domain is tested. If any of these tests fail, it results in an e-mail and a page for the admins.
- Swatch runs constantly on our central log server monitoring logs for certain key system events. If any of these events are detected in the logs, the admins are notified via e-mail virtually immediately.
- A script called check_processes.sh[80] runs from cron every 5 minutes that, you guessed it, checks to see if certain processes are running. If not, the script starts the process in question and generates an e-mail for the admins.

[78] Tripwire must have a database to compare against so the file information database must be created first. This action will create a file called "tw.db_[hostname]" in the directory specified to hold your databases.
[79] It is also a good idea to make a hardcopy printout of the database contents right away. In the event that you become suspicious of the integrity of the database, you will be able to manually compare information against this hardcopy.

Integrity Reports, Log Monitoring, and Security Alert Notifications -- Daily

- The tripwire configuration outlined in step 9 of the step by step will produce a daily disk integrity report. This report will be evaluated daily and any discrepancies will be dealt with accordingly. This includes investigating any strange results and/or updating the tripwire database as necessary.
- A script runs daily on the central log server that parses yesterday's logs into date-named files and puts them in server-named directories. These files are sorted by syslog facility and then by time, for easy perusal. After this job runs, a quick-and-dirty log review will be performed daily.
- Subscribing to a few well-written Security Alert Notification services can be an administrator's best friend. I personally subscribe to CERT advisories, the SANS Security Alert Consensus mailing list, and the Sun Security mailing list, to name a few. I also frequent the Internet Storm Center and other vendor sites (isc.org, openssh.org, etc.) enough to classify this type of work as an every day thing.
- A script runs nightly on this remote slave server that performs a few simple tasks to satisfy admin paranoia: it concatenates the output of netstat –a, who –T, and ps –ef into a file and sends an e-mail. This "snapshot" simply provides a quick glimpse at what's going on after hours on a given server.

Backups -- Weekly

- Since this box is external to our network and our facilities here, applying our normal corporate backup solution to this server is not really feasible, nor appropriate. Since this machine really only contains a small amount of unique information (mostly in the form of configuration files and logs), weekly backups of a tar variety will suffice. And since the amount of data involved is relatively small, a simple script[81] is used to scp the date-named archive back to our corporate network. These archives are kept in server-named directories on a volume here that is in turn backed up to tape. The comments in the script say nothing to this effect, but ns2's public key has been copied to the archive server on our network so that scp can be non-interactive.

Patch Review -- Monthly

- Once a month it's a good idea (if you've got the time!) to review the patch level of a box. The end result of this review may or may not result in the application of new patches and/or software upgrades – the important thing is that the OS and any applications installed have been reviewed. The mailing lists referred to above assist greatly in this task, acting as constant reminders to keep your act together in the patching arena. A monthly patchdiag against the most recent patchdiag.xref is always informative, especially when compared against the results of the previous patchdiag output. And finally, visits to all vendor web sites looking for application or security related patches are a must. Sites visited include sunsolve.sun.com, isc.org, openssh.org, openssl.org, and gzip.org/zlib.

---

[80] An example of check_processes.sh can be found in Appendix Q.
[81] An example of backup_slave.sh can be found in Appendix R.

Periodic Scans -- Every 3 Months

- Nmap is a very useful tool – it's a free port scanner that can scan single IPs or entire networks. In addition, it can scan all TCP ports, all UDP ports, and any applications running RPC. It can even make a guess at the remote OS type! Its reports are clear and concise and come in a variety of formats. It is a simple task to run this tool from our security server here and review the report.
- Nessus is good tool as well – it's a free security-checking tool that tests for a slew of known vulnerabilities by running various homegrown algorithms. Making sure the plug-ins are up to date is useful for getting the Nessus approach to the most current vulnerabilities circulating. It has a nice GUI and its reports not only provide useful information about the exploit you are vulnerable to, but they also provide feedback on how to deal with them. Again, it's a simple task to run this tool from our security server here and review the reports. It should be noted that both these scans can be fairly traffic-intensive, so off-peak hours is when we schedule such tests.

➢ **Last Phase: Verify and Test Configurations**

After a machine has been fully hardened and secured, tests should be run to ensure that the steps taken to secure the box were done thoroughly and completely. The following tests were performed as a means to verify that the box is in fact, configured correctly. They were also designed to possibly uncover some aspect of securing the server that may have been overlooked.

Test 1 -- Port Scans, Connections, and Processes

- One of the biggest priorities for securing this server is making sure that only two ports are open: port 53 tcp/udp for DNS services and port 22 tcp for ssh. Nmap is run from a security server against ns2 to discover which ports are in fact open. The command used for this:

  nmap –sT –sU –sR –P0 –O –oN /tmp/nmap_scan_ns2.out[82] ns2

  –sT enables tcp port scans
  –sU enables udp port scans
  –sR enables RPC scans
  –P0 disables ping tests
  –O attempts to guess the OS
  –oN specifies Normal output to file /tmp/nmap_scan_ns2.out

- Running the command /bin/netstat –a from ns2 will display all active sockets, both udp and tcp. This should corroborate the nmap results.
- Finally, run /bin/ps –ef to get a list of running processes. Any process appearing in this list that cannot be verified as a necessary process should be killed and disabled.

[82] The results of Test 1 can be reviewed in Appendix S.

Test 2 -- Alerts[83]

- Logging for ns2 has been configured through syslogd to log both locally and remotely to a centralized log server. Swatch has been configured on this log server to run against the incoming logs from ns2, and alert the admins when certain system events occur. A sampling of these system events should be simulated to see what alerts Swatch will generate.
- If the /etc/hosts.allow was configured correctly during Test 1 above, then an alert for ssh should have been generated during the nmap scan.
- SiteScope has been configured to run several DNS tests against ns2, and has been set to alert via both pager and e-mail. Two random zones served by this DNS slave server were unloaded and alerts were generated. Then named was shutdown altogether, and all DNS alerts were triggered.

Test 3 -- BIND security[84]

- Zone transfers can be configured to be significantly more secure in BIND9, and that is what we have attempted to do in this server's configuration. Test that security by attempting an unauthorized zone transfer.
- Then verify that zone transfers will work for a legitimate request.
- Views are new in BIND9 and provide a level of DNS security by obscurity. Often, large domains contain some DNS entries meant for private internal use only (i.e. hosts on the private network) as well as DNS entries meant only for the outside world (externally routable IPs). The outside world should never see these private entries and views allow you to configure this, which was a goal of ns2's BIND configuration. Test this by issuing several queries from a variety of hosts.

Test 4 -- Tripwire

- Host-based intrusion detection tools like Tripwire are a must for any server that is going to be visible to the Internet. The proliferation of root kits, trojans, and chat relays is such that no server is "unhackable". Accepting this as a reality of the job, if an admin can't prevent all intrusions perhaps it is enough just to know when and to what extent the breach occurred. Tripwire is a great tool for this when properly configured.
- Test the Tripwire configuration by making some changes to various systems files and then run the report. This report should corroborate all the changes made.
- Add a user with uid 0 by manually editing the /etc/passwd and /etc/shadow files.
- Replace the Solaris tar binary with the GNU tar binary.
- Add a directory under /dev named lp.
- Delete the /etc/ftpusers file.
- Run the integrity report[85]: /opt/tw/bin/tw/tripwire

[83] The results of Test 2 can be reviewed in Appendix T.
[84] The results of Test 3, along with some security comments, can be reviewed in Appendix U.
[85] The results of Test 4 can be reviewed in Appendix V.

Test 5 -- Security Scanners

- Remote security scanners are a great idea and they are starting to pop up everywhere. Unfortunately, most of these products are commercial, and fairly expensive at that. Nessus is a remote security scanner that is free, powerful, and easy to use. It requires a client build on ns2, which is nothing more than some libraries, a few utilities, and nessusd. Our corporate security server already has an installation of the nessus server and can connect to any properly configured server running nessusd. This nessus scan will be performed by connecting to ns2 from our security server and running the full gamut of plug-in tests.
- On ns2: /usr/local/sbin/nessusd –D &[86]
- From security server: /usr/local/bin/nessus &
- Use the server-side gui to log into ns2.
- Perform all tests, scans, and attacks accept DOS attacks.
- Accept most of the defaults.
- Specify ns2 as the target and attempt a zone transfer.
- Start the scan and review the report[87] once the scan completes.

[86] nessusd should not be configured to start automatically at boot, but should instead be run manually like this whenever a scan is required. Because nessusd must be running to generate a scan report, tcp port 1241 shows up in the results as an open port. Ignore this, as nessusd will not usually be running.
[87] The results of Test 5 can be reviewed in Appendix W.

Appendix A: Patchdiag output

```
================================================================================
System Name: ns2   SunOS Vers: 5.8      Arch: sparc
Cross Reference File Date: Sep/02/02

PatchDiag Version: 1.0.4
================================================================================
Report Note:

Recommended patches are considered the most important and highly
recommended patches that avoid the most critical system, user, or
security related bugs which have been reported and fixed to date.
A patch not listed on the recommended list does not imply that it
should not be used if needed.  Some patches listed in this report
may have certain platform specific or application specific dependencies
and thus may not be applicable to your system.  It is important to
carefully review the README file of each patch to fully determine
the applicability of any patch with your system.
================================================================================
INSTALLED PATCHES
Patch  Installed Latest   Synopsis
  ID   Revision  Revision
------ --------- -------- ------------------------------------------------------------------------------------------------------------
108528    15     CURRENT  SunOS 5.8: kernel update patch
108723    01     CURRENT  SunOS 5.8: /kernel/fs/lofs and /kernel/fs/sparcv9/lofs patch
108725    09     CURRENT  SunOS 5.8: st driver patch
108727    16     CURRENT  SunOS 5.8: /kernel/fs/nfs and /kernel/fs/sparcv9/nfs patch
108806    12     CURRENT  SunOS 5.8: Sun Quad FastEthernet qfe driver
108813    06       10     SunOS 5.8: Sun Gigabit Ethernet 3.0
108820    01     CURRENT  SunOS 5.8: nss_compat.so.1 patch
108823    01     CURRENT  SunOS 5.8: compress/uncompress/zcat patch
108825    01     CURRENT  SunOS 5.8: /usr/lib/fs/cachefs/cfsadmin patch
108827    26     CURRENT  SunOS 5.8: /usr/lib/libthread.so.1 patch
108875    12     CURRENT  SunOS 5.8: c2audit patch
108899    01     CURRENT  SunOS 5.8: /usr/bin/ftp patch
108901    05     CURRENT  SunOS 5.8: /kernel/sys/rpcmod and /kernel/strmod/rpcmod patch
108914    02     CURRENT  SunOS 5.8: l10n update: PDA Sync, SmartCard, DHCP mgr, Printer Adm
108954    02     CURRENT  SunOS 5.8: localization updates for different components
108964    05       06     SunOS 5.8: /usr/sbin/in.tftpd and /usr/sbin/snoop patch
108968    07     CURRENT  SunOS 5.8: vol/vold/rmmount/dev_pcmem.so.1 patch
108970    01     CURRENT  SunOS 5.8: /usr/lib/fs/pcfs/fsck and /usr/lib/fs/pcfs/mkfs patch
108972    04     CURRENT  SunOS 5.8: /sbin/fdisk patch
108974    23     CURRENT  SunOS 5.8: dada, uata, dad, sd and scsi drivers patch
108975    06     CURRENT  SunOS 5.8: /usr/bin/rmformat and /usr/sbin/format patch
108977    01     CURRENT  SunOS 5.8: libsmedia patch
108981    08     CURRENT  SunOS 5.8: /kernel/drv/hme and /kernel/drv/sparcv9/hme patch
108982    09     CURRENT  SunOS 5.8: fctl/fp/fcp/usoc driver patch
108983    08     CURRENT  SunOS 5.8: /kernel/drv/fcip driver patch
108984    08     CURRENT  SunOS 5.8: /kernel/drv/qlc driver patch
108985    03     CURRENT  SunOS 5.8: /usr/sbin/in.rshd patch
108987    09     CURRENT  SunOS 5.8: Patch for patchadd and patchrm
108989    02     CURRENT  SunOS 5.8: /usr/kernel/sys/acctctl and /usr/kernel/sys/exacctsys p
108991    13       18     Obsoleted by: 108827-15 SunOS 5.8: /usr/lib/libc.so.1 patch
108993    11     CURRENT  SunOS 5.8: nss and ldap patch
108995    02       03     SunOS 5.8: /usr/lib/libproc.so.1 patch
```

108997 03 CURRENT SunOS 5.8: libexacct and libproject patch
108999 01 CURRENT SunOS 5.8: PAM patch
109003 01 CURRENT SunOS 5.8: /etc/init.d/acctadm and /usr/sbin/acctadm patch
109005 02 03 SunOS 5.8: /sbin/su.static and /usr/bin/su patch
109007 07 CURRENT SunOS 5.8: at/atrm/batch/cron patch
109009 01 02 SunOS 5.8: /etc/magic and /usr/bin/file patch
109011 01 CURRENT SunOS 5.8: /usr/bin/id and /usr/xpg4/bin/id patch
109013 02 CURRENT SunOS 5.8: /usr/bin/lastcomm patch
109015 01 CURRENT SunOS 5.8: /usr/bin/newtask patch
109017 01 CURRENT SunOS 5.8: /usr/bin/pgrep and /usr/bin/pkill patch
109019 02 CURRENT SunOS 5.8: /usr/bin/priocntl patch
109021 01 CURRENT SunOS 5.8: /usr/bin/projects patch
109023 01 CURRENT SunOS 5.8: /usr/bin/sparcv7/ps and /usr/bin/sparcv9/ps patch
109025 03 04 SunOS 5.8: /usr/bin/sparcv7/truss and /usr/bin/sparcv9/truss patch
109027 01 CURRENT SunOS 5.8: /usr/bin/wracct patch
109029 02 CURRENT SunOS 5.8: perl patch
109031 01 CURRENT SunOS 5.8: projadd/projdel/projmod patch
109033 01 CURRENT SunOS 5.8: /usr/bin/sparcv7/prstat and /usr/bin/sparcv9/prstat pat
109035 02 CURRENT SunOS 5.8: useradd/userdel/usermod patch
109037 01 CURRENT SunOS 5.8: /var/yp/Makefile and /var/yp/nicknames patch
109043 02 CURRENT SunOS 5.8: sonode adb macro patch
109045 02 03 SunOS 5.8: /usr/sbin/sparcv7/crash and /usr/sbin/sparcv9/crash pat
109077 04 10 SunOS 5.8: dhcp server and admin patch
109091 05 CURRENT SunOS 5.8: /usr/lib/fs/ufs/ufsrestore patch
109145 01 CURRENT SunOS 5.8: /usr/sbin/in.routed patch
109147 18 CURRENT SunOS 5.8: Linker patch
109149 01 02 SunOS 5.8:: /usr/sbin/mkdevmaps and /usr/sbin/mkdevalloc patch
109181 04 CURRENT Obsoleted by: 108528-13 SunOS 5.8: /kernel/fs/cachefs patch
109202 01 03 SunOS 5.8: /kernel/misc/gld and /kernel/misc/sparcv9/gld patch
109223 02 CURRENT SunOS 5.8: kpasswd, libgss.so.1 and libkadm5clnt.so.1 patch
109234 09 CURRENT SunOS 5.8: Apache Security and NCA Patch
109238 02 CURRENT SunOS 5.8: /usr/bin/sparcv7/ipcs and /usr/bin/sparcv9/ipcs patch
109277 02 CURRENT SunOS 5.8: /usr/bin/iostat patch
109279 14 18 Obsoleted by: 108528-13 SunOS 5.8: /kernel/drv/ip patch
109318 18 28 SunOS 5.8: suninstall patch
109322 07 09 Obsoleted by: 108827-15 SunOS 5.8: libnsl patch
109324 04 CURRENT SunOS 5.8: sh/jsh/rsh/pfsh patch
109326 09 CURRENT SunOS 5.8: libresolv.so.2 and in.named patch
109328 01 02 SunOS 5.8: /usr/lib/netsvc/yp/ypserv and usr/lib/netsvc/yp/ypxfr p
109384 01 04 SunOS 5.8: libaio patch
109454 01 CURRENT SunOS 5.8: /kernel/fs/fifofs and /kernel/fs/sparcv9/fifofs patch
109458 02 CURRENT SunOS 5.8: /kernel/strmod/ldterm patch
109472 06 07 Obsoleted by: 108528-13 SunOS 5.8: /kernel/drv/tcp patch
109524 05 13 SunOS 5.8: /kernel/drv/ssd patch
109529 06 CURRENT SunOS 5.8: luxadm, liba5k and libg_fc patch
109576 01 CURRENT SunOS 5.8: mountall and fsckall patch
109657 07 CURRENT SunOS 5.8: isp driver patch
109667 04 CURRENT SunOS 5.8: /usr/lib/inet/xntpd and /usr/sbin/ntpdate patch
109729 01 CURRENT SunOS 5.8: /usr/bin/cat patch
109740 04 CURRENT Obsoleted by: 108528-13 SunOS 5.8: /kernel/drv/udp patch
109742 04 CURRENT Obsoleted by: 108528-13 SunOS 5.8: /kernel/drv/icmp patch
109764 02 04 SunOS 5.8: /kernel/fs/hsfs and /kernel/fs/sparcv9/hsfs patch
109783 01 CURRENT SunOS 5.8: /usr/lib/nfs/nfsd patch
109785 01 CURRENT SunOS 5.8: /etc/inittab patch
109793 11 CURRENT SunOS 5.8: su driver patch
109803 01 CURRENT SunOS 5.8: /usr/bin/du and /usr/xpg4/bin/du patch

109805 11 CURRENT SunOS 5.8: /usr/lib/security/pam_krb5.so.1 patch
109807 01 CURRENT SunOS 5.8: /usr/sbin/dumpadm patch
109809 01 CURRENT SunOS 5.8: timezone data patch for Australasia
109815 07 14 SunOS 5.8: se, acebus, pcf8574, pcf8591 and scsb patch
109874 06 CURRENT Obsoleted by: 109896-07 SunOS 5.8: audio patch
109876 02 CURRENT SunOS 5.8: fd driver patch
109877 01 CURRENT SunOS 5.8: /usr/include/sys/dma_i8237A.h patch
109879 02 CURRENT SunOS 5.8: isadma driver patch
109882 06 CURRENT SunOS 5.8: eri header files patch
109883 02 CURRENT SunOS 5.8: /usr/include/sys/ecppsys.h patch
109885 09 CURRENT SunOS 5.8: glm patch
109888 15 CURRENT SunOS 5.8: platform drivers patch
109893 02 03 SunOS 5.8: stc driver patch
109894 01 CURRENT SunOS 5.8: /kernel/drv/sparcv9/bpp driver patch
109896 04 08 SunOS 5.8: USB and Audio Framework patch
109898 05 CURRENT SunOS 5.8: /kernel/drv/arp patch
109900 02 CURRENT SunOS 5.8: /etc/init.d/network and /sbin/ifparse patch
109902 03 CURRENT SunOS 5.8: /usr/lib/inet/in.ndpd patch
109904 04 05 Obsoleted by: 108528-13 SunOS 5.8: /etc/default/mpathd and /sbin/i
109906 06 CURRENT Obsoleted by: 108528-13 SunOS 5.8: dhcpagent, dhcpinfo, ifconfig a
109920 05 07 SunOS 5.8: pcic driver patch
109922 02 03 SunOS 5.8: pcelx and pcser driver patch
109924 02 04 SunOS 5.8: pcata driver patch
109926 02 CURRENT SunOS 5.8: /kernel/drv/pem and /kernel/drv/sparcv9/pem patch
109928 04 05 SunOS 5.8: pcmem and pcmcia patch
109933 01 CURRENT SunOS 5.8: mv, cp, ln patch
109936 01 CURRENT SunOS 5.8: /usr/bin/diff patch
109954 01 CURRENT Obsoleted by: 108528-13 SunOS 5.8: /kernel/sys/pset and /kernel/sy
109994 01 CURRENT SunOS 5.8: /usr/bin/sparcv7/adb and /usr/bin/sparcv9/adb patch
110075 01 CURRENT SunOS 5.8: /kernel/drv/devinfo and /kernel/drv/sparcv9/devinfo pat
110165 02 03 SunOS 5.8: /usr/bin/sed patch
110269 01 CURRENT SunOS 5.8: /usr/lib/libnisdb.so.2 patch
110274 03 CURRENT SunOS 5.8: Figgs Custom install new features and install help
110283 05 CURRENT SunOS 5.8: mkfs and newfs patch
110322 02 CURRENT SunOS 5.8: /usr/lib/netsvc/yp/ypbind patch
110368 02 CURRENT SunOS 5.8: pcf8574 driver patch for SUNW Sun-Fire-280R
110369 04 05 SunOS 5.8: sgcn patch
110371 02 03 SunOS 5.8: serengeti support, Update3, sgfru patch
110373 02 04 SunOS 5.8: /platform/SUNW,Sun-Fire/kernel/drv/sparcv9/sgsbbc patch
110374 06 08 SunOS 5.8: /platform/SUNW,Sun-Fire/kernel/drv/sparcv9/sgenv patch
110379 01 CURRENT SunOS 5.8: littleneck support, gpio patch
110380 04 CURRENT SunOS 5.8: ufssnapshots support, libadm patch
110381 01 CURRENT SunOS 5.8: ufssnapshots support, clri patch
110382 01 CURRENT SunOS 5.8: file systems should support snapshots for online bkups
110383 02 CURRENT Obsoleted by: 108528-13 SunOS 5.8: libnvpair patch
110384 05 CURRENT Obsoleted by: 108528-11 SunOS 5.8: RCM libraries & header patch
110385 03 CURRENT SunOS 5.8: RCM modules patch
110386 01 02 SunOS 5.8: RBAC Feature Patch
110387 03 CURRENT SunOS 5.8: ufssnapshots support, ufsdump patch
110388 01 CURRENT SunOS 5.8: RBAC Feature for Solaris Update 3
110390 02 CURRENT Obsoleted by: 108993-05 SunOS 5.8: ldapclient patch
110458 02 CURRENT SunOS 5.8: libcurses patch
110460 20 CURRENT SunOS 5.8: fruid/PICL plug-ins patch
110461 01 CURRENT SunOS 5.8: ttcompat patch
110511 01 04 SunOS 5.8: rpc.nisd patch
110609 02 CURRENT SunOS 5.8: cdio.h and command.h USB header patch

```
110615 05    CURRENT  SunOS 5.8: sendmail patch
110662 07    CURRENT  SunOS 5.8: ksh patch
110668 03    CURRENT  SunOS 5.8: /usr/sbin/in.telnetd patch
110700 01    CURRENT  SunOS 5.8: automount patch
110716 02    CURRENT  SunOS 5.8: Solaris Product Registry 3.0 patch
110797 02    CURRENT  SunOS 5.8: UR4 New int
110811 01    CURRENT  SunOS 5.8: libnls patch
110815 01    CURRENT  SunOS 5.8:  libmp patch
110820 03      08     SunOS 5.8: /platform/SUNW,Sun-Fire-15000/kernel/drv/sparcv9/dman p
110821 02    CURRENT  SunOS 5.8: iosram driver patch
110826 03      05     SunOS 5.8: SUNW,Sun-Fire-15000/kernel/drv/sparcv9/schpc patch
110828 01      02     SunOS 5.8: sbbc driver patch
110838 05    CURRENT  SunOS 5.8: /platform/SUNW,Sun-Fire-15000/kernel/drv/sparcv9/axq pa
110839 01      03     SunOS 5.8: /usr/lib/rcm/modules/SUNW_ip_rcm.so patch
110840 02    CURRENT  SunOS 5.8: bbc patch
110841 01    CURRENT  SunOS 5.8: gptwo patch
110842 08    CURRENT  SunOS 5.8: hpc3130 driver patch for SUNW,Sun-Fire-880
110844 02    CURRENT  SunOS 5.8: /platform/sun4u/kernel/drv/sparcv9/lm75 patch
110845 03    CURRENT  SunOS 5.8: /platform/sun4u/kernel/drv/sparcv9/ltc1427 patch
110847 02    CURRENT  SunOS 5.8: /platform/sun4u/kernel/drv/sparcv9/pcf8591 patch
110851 01      02     SunOS 5.8: /platform/sun4u/kernel/drv/sparcv9/ssc050 patch
110852 02      03     SunOS 5.8: /platform/sun4u/kernel/drv/sparcv9/ssc100 patch
110854 02    CURRENT  SunOS 5.8: /platform/sun4u/kernel/drv/sparcv9/smbus_ara patch
110856 01    CURRENT  SunOS 5.8: /etc/inet/services patch
110888 01    CURRENT  SunOS 5.8 : figgs, New and updated message strings
110896 01    CURRENT  SunOS 5.8: /usr/lib/fs/cachefs/mount patch
110898 04    CURRENT  SunOS 5.8: csh/pfcsh patch
110901 01    CURRENT  SunOS 5.8: /kernel/drv/sgen and /kernel/drv/sparcv9/sgen patch
110903 05    CURRENT  SunOS 5.8: edit, ex, vedit, vi and view patch
110905 01      02     SunOS 5.8: /usr/bin/find patch
110910 01    CURRENT  SunOS 5.8: /usr/lib/fs/ufs/fsck patch
110912 01      03     SunOS 5.8: cfgadm patch
110914 01    CURRENT  SunOS 5.8: /usr/bin/tr patch
110916 03    CURRENT  SunOS 5.8: sort patch
110918 01      03     SunOS 5.8: /kernel/drv/openeepr patch
110934 08    CURRENT  SunOS 5.8: pkgtrans, pkgadd, pkgchk and libpkg.a patch
110945 06    CURRENT  SunOS 5.8: /usr/sbin/syslogd patch
110949 01    CURRENT  Obsoleted by: 110934-04 SunOS 5.8: /usr/sadm/install/bin/pkgremove
110951 02    CURRENT  SunOS 5.8: /usr/sbin/tar and /usr/sbin/static/tar patch
110955 01      03     SunOS 5.8: /kernel/strmod/timod patch
110957 02    CURRENT  SunOS 5.8: /usr/bin/mailx patch
111016 01    CURRENT  SunOS 5.8: /usr/bin/sdiff patch
111018 01    CURRENT  SunOS 5.8: /etc/driver_aliases patch for gpio
111019 06
111021 03
111023 01    CURRENT  SunOS 5.8: /kernel/fs/mntfs and /kernel/fs/sparcv9/mntfs patch
111035 01    CURRENT  Obsoleted by: 108528-13 SunOS 5.8: /kernel/sys/doorfs and /kernel/
111069 01    CURRENT  SunOS 5.8: bsmunconv overwrites root cron tab if cu created /tmp/r
111085 02    CURRENT  SunOS 5.8: /usr/bin/login patch
111090 01      03     Obsoleted by: 108993-05 SunOS 5.8: /usr/lib/libsldap.so.1 patch
111098 01    CURRENT  SunOS 5.8: ROC timezone should be avoided for political reasons
111111 03    CURRENT  SunOS 5.8: /usr/bin/nawk patch
111141 01    CURRENT  SunOS 5.8: last works incorrectly for more than 256 users login
111177 03      06     Obsoleted by: 108827-15 SunOS 5.8: /usr/lib/lwp/libthread.so.1 pat
111197 01      02     SunOS 5.8: /usr/lib/nfs/mountd patch
111225 01      02     SunOS 5.8: /usr/bin/tail and /usr/xpg4/bin/tail patch
```

111232 01 CURRENT SunOS 5.8: patch in.fingerd
111234 01 CURRENT SunOS 5.8: patch finger
111265 01 CURRENT SunOS 5.8: patch who
111267 02 CURRENT Obsoleted by: 111588-02 SunOS 5.8: /kernel/fs/specfs patch
111275 01 CURRENT SunOS 5.8: New features Solaris 8 Update 5 European
111293 04 CURRENT SunOS 5.8: /usr/lib/libdevinfo.so.1 patch
111295 01 CURRENT SunOS 5.8: /usr/bin/sparcv7/pstack & /usr/bin/sparcv9/pstack patch
111297 01 CURRENT SunOS 5.8: /usr/lib/libsendfile.so.1 patch
111299 01 03 SunOS 5.8: PPP patch
111302 01 CURRENT SunOS 5.8: EDHCP libraries patch
111304 01 CURRENT SunOS 5.8: /kernel/misc/nfs_dlboot patch
111306 01 03 SunOS 5.8: ufsboot and inetboot patch
111308 01 02 SunOS 5.8: /usr/lib/libmtmalloc.so.1 patch
111310 01 CURRENT SunOS 5.8: /usr/lib/libdhcpagent.so.1 patch
111317 01 02 SunOS 5.8: /sbin/init and /usr/sbin/init patch
111319 01 CURRENT SunOS 5.8: /usr/sbin/in.rdisc patch
111321 01 02 SunOS 5.8: klmmod and klmops patch
111325 02 CURRENT SunOS 5.8: /usr/lib/saf/ttymon patch
111327 05 CURRENT SunOS 5.8: libsocket patch
111363 01 CURRENT Obsoleted by: 110934-04 SunOS 5.8: /usr/sbin/installf patch
111368 01 CURRENT SunOS 5.8: /usr/bin/groups patch
111504 01 CURRENT SunOS 5.8: /usr/bin/tip patch
111548 01 CURRENT SunOS 5.8: catman, man, whatis, apropos and makewhatis patch
111606 02 CURRENT SunOS 5.8: /usr/sbin/in.ftpd patch
111659 07 CURRENT SunOS 5.8: passwd and pam_unix.so.1 patch
111826 01 CURRENT SunOS 5.8: /usr/sbin/sparcv7/whodo & /usr/sbin/sparcv9/whodo patch
111874 05 CURRENT SunOS 5.8: usr/bin/mail patch
111881 02 CURRENT SunOS 5.8: /usr/kernel/strmod/telmod patch
111958 02 CURRENT SunOS 5.8: /usr/lib/nfs/statd patch
112138 01 CURRENT SunOS 5.8:: usr/bin/domainname patch
112218 01 CURRENT SunOS 5.8:: pam_ldap.so.1 patch
112237 05 CURRENT SunOS 5.8: mech_krb5.so.1 patch
112254 01 CURRENT SunOS 5.8: /kernel/sched/TS patch
112325 01 CURRENT SunOS 5.8: /kernel/fs/udfs and /kernel/fs/sparcv9/udfs patch
112396 02 CURRENT SunOS 5.8: /usr/bin/fgrep patch
112425 01 CURRENT SunOS 5.8: /usr/lib/fs/ufs/mount and /etc/fs/ufs/mount patch
112459 01 CURRENT SunOS 5.8: /usr/lib/pt_chmod patch
112796 01 CURRENT SunOS 5.8: /usr/sbin/in.talkd patch
112846 01 CURRENT SunOS 5.8: /usr/lib/netsvc/rwall/rpc.rwalld patch

==========

UNINSTALLED RECOMMENDED PATCHES

Patch ID	Ins Rev	Lat Rev	Age	Require ID	Incomp ID	Synopsis
108652	N/A	56	38			X11 6.4.1: Xsun patch
108869	N/A	18	72			SunOS 5.8: snmpdx/mibiisa/libssasnmp/snmplib patch
108919	N/A	15	85	108652-19		CDE 1.4: dtlogin patch
108949	N/A	07	272			CDE 1.4: libDtHelp/libDtSvc patch
109041	N/A	04	472	108528-08		Obsoleted by: 108528-09 SunOS 5.8: sockfs patch
109137	N/A	01	883			Obsoleted by: 110934-03 SunOS 5.8: /usr/sadm/install/bin/pkginstal
109221	N/A	06	587	108993-01		Obsoleted by: 109318-12 SunOS 5.8: Patch for sysidnet
109320	N/A	05	156			SunOS 5.8: LP patch
109470	N/A	02	737			CDE 1.4: Actions Patch
109587	N/A	03	448			Obsoleted by: 109318-18 SunOS 5.8: libspmistore patch

```
109951 N/A  01 749                SunOS 5.8: jserver buffer overflow
110286 N/A  09   7                OpenWindows 3.6.2: Tooltalk patch
110453 N/A  03 231                SunOS 5.8: admintool patch
110670 N/A  01 524                SunOS 5.8: usr/sbin/static/rcp patch
110723 N/A  05  92 109882-06          SunOS 5.8: /kernel/drv/sparcv9/eri patch
110939 N/A  01 548                SunOS 5.8: /usr/lib/acct/closewtmp patch
110943 N/A  01 469                SunOS 5.8: /usr/bin/tcsh patch
111071 N/A  01 524                SunOS 5.8: cu patch
111570 N/A  01 435                SunOS 5.8: uucp patch
111596 N/A  02 373 111659-01          SunOS 5.8: /usr/lib/netsvc/yp/rpc.yppasswdd patch
111626 N/A  02  30                OpenWindows 3.6.2: Xview Patch
111879 N/A  01 374                SunOS 5.8: Solaris Product Registry patch SUNWwsr
112279 N/A  02  97                SunOS 5.8: pkgrm failed during upgrade from Solaris 8 to Solaris 9
112334 N/A  02 184                Obsoleted by: 108528-14 SunOS 5.8: /usr/include/sys/archsystm.h pa
112611 N/A  01 142                SunOS 5.8: /usr/lib/libz.so.1 patch
112668 N/A  01 114                SunOS 5.8: /usr/bin/gzip patch
=============================================================================
```

UNINSTALLED SECURITY PATCHES

NOTE: This list includes the Security patches that are also Recommended

```
Patch  Ins Lat Age  Require    Incomp  Synopsis
  ID   Rev Rev        ID         ID
------ --- --- --- --------- --------- ----------------------------------------------------------------------------------------------------
108652 N/A  56  38                X11 6.4.1: Xsun patch
108773 N/A  15  43                SunOS 5.8: IIIM and X Input & Output Method patch
108835 N/A  03 154                CDE 1.4: dtcm patch
108869 N/A  18  72                SunOS 5.8: snmpdx/mibiisa/libssasnmp/snmplib patch
108909 N/A  12 301 109887-09          CDE 1.4: Smart Card Administration GUI patch
108949 N/A  07 272                CDE 1.4: libDtHelp/libDtSvc patch
108979 N/A  10 659 108528-03          Obsoleted by: 108528-04 SunOS 5.8: platform nexus, I2C, Netra ct
a
109041 N/A  04 472 108528-08          Obsoleted by: 108528-09 SunOS 5.8: sockfs patch
109134 N/A  27  16 109318-06          SunOS 5.8: WBEM patch
               110386-01
109154 N/A  14  38                SunOS 5.8: PGX32 Graphics
109320 N/A  05 156                SunOS 5.8: LP patch
109354 N/A  14 220 108652-19          CDE 1.4: dtsession patch
109695 N/A  03 409                SunOS 5.8: /etc/smartcard/opencard.properties patch
109887 N/A  13 132 108528-14          SunOS 5.8: smartcard and usr/sbin/ocfserv patch
109892 N/A  03 555 108528-06          SunOS 5.8: /kernel/drv/ecpp driver patch
               109877-01
               109883-01
109951 N/A  01 749                SunOS 5.8: jserver buffer overflow
109965 N/A  03 587                Obsoleted by: 109887-02 SunOS 5.8: pam_smartcard.so.1 patch
110068 N/A  02 527                CDE 1.4: PDASync patch
110286 N/A  09   7                OpenWindows 3.6.2: Tooltalk patch
110416 N/A  03 399                SunOS 5.8: ATOK12 patch
110453 N/A  03 231                SunOS 5.8: admintool patch
110670 N/A  01 524                SunOS 5.8: usr/sbin/static/rcp patch
110943 N/A  01 469                SunOS 5.8: /usr/bin/tcsh patch
111071 N/A  01 524                SunOS 5.8: cu patch
111332 N/A  05 150                SunOS 5.8: /usr/lib/dcs patch
111400 N/A  01 442                SunOS 5.8: KCMS configure tool has a security vulnerability
111570 N/A  01 435                SunOS 5.8: uucp patch
```

```
111596 N/A  02 373 111659-01          SunOS 5.8: /usr/lib/netsvc/yp/rpc.yppasswdd patch
111624 N/A  03 148                   SunOS 5.8: /usr/sbin/inetd patch
111626 N/A  02  30                   OpenWindows 3.6.2: Xview Patch
111647 N/A  01 395                   BCP libmle buffer overflow
112039 N/A  01 353                   SunOS 5.8: usr/bin/ckitem patch
112390 N/A  05  52 109223-02         SunOS 5.8: Supplemental Encryption Kerberos V5:
mech_krb5.so.1 pat
112438 N/A  01 161                   SunOS 5.8: /kernel/drv/random patch
112605 N/A  03  70                   SunOS 5.8: /kernel/fs/autofs and /kernel/fs/sparcv9/autofs patch
112611 N/A  01 142                   SunOS 5.8: /usr/lib/libz.so.1 patch
112668 N/A  01 114                   SunOS 5.8: /usr/bin/gzip patch
112792 N/A  01  58 108968-06         SunOS 5.8: /usr/lib/pcmciad patch
================================================================================

UNINSTALLED Y2K PATCHES

NOTE: This list includes the Y2K patches that are also Recommended

Patch  Ins Lat Age  Require    Incomp  Synopsis
  ID   Rev Rev        ID         ID
------ --- --- --- --------- --------- --------------------------------------------------------------------------------
All Y2K patches installed!
================================================================================

OTHER RELATED UNINSTALLED PATCHES

NOTE: This is determined by the packages that have been
      installed on the system.

      When one patch refers to multiple packages, we list the
      additional packages in the next lines.

      The various 'S','R','*' marks denote unbundled packages
      that is designated as an 'Security' or 'Recommended'.

        S = Security
        R = Recommened Unbundled
        * = Both Security and Recommended Unbundled

Patch    Package   Lat Age Synopsis
  ID     Name      Rev
------ - --------- --- --- --------------------------------------------------------------------------------
108874   SUNWcar    01 903 Obsoleted by: 108528-04 SunOS 5.8: fhc driver patch
108966   SUNWcsr    06 657 Obsoleted by: 108528-05 SunOS 5.8: /kernel/fs/ufs and /kernel/fs/s
109236   SUNWcsr    01 595 Obsoleted by: 108528-05 SunOS 5.8: msgsys, semsys and shmsys patch
109461   SUNWcsl    03 482 Obsoleted by: 111177-02 SunOS 5.8: /usr/lib/lwp/libthread.so.1 pat
109571   SUNWcsu    02 766 Obsoleted by: 108528-05 SunOS 5.8: /usr/sbin/devfsadm patch
109680   SUNWcsl    01 667 Obsoleted by: 108991-12 SunOS 5.8: nss_nisplus.so.1 and libnss_nis
109801   SUNWcsl    01 764 Obsoleted by: 108528-05 SunOS 5.8: libdevice.so.1 patch
109872   SUNWcar    01 665 SunOS 5.8: vis driver patch
109880   SUNWcar    01 680 Obsoleted by: 108528-04 SunOS 5.8: forthdebug patch
109889   SUNWkvm    01 596 SunOS 5.8: usr platform links and libc_psr patch
110285   SUNWcar    01 521 SunOS 5.8: consconfig_dacf patch
110370   SUNWkvm    03 295 SunOS 5.8: SUNW,Sun-Fire usr platform links patch
110372   SUNWcsr    02 521 Obsoleted by: 108528-07 SunOS 5.8: serengeti support, Update3, sgh
110376   SUNWkvm    01 521 SunOS 5.8: littleneck support, usr_platform patch, S8 Update 3
```

110614 SUNWses 02 443 SunOS 5.8: ses driver patch
110692 SUNWqfed 03 541 NSS 1.0: patch for Netra Software Suite Network Resilience
110710 SUNWcsu 01 358 SunOS 5.8: nscd patch
110833 SUNWkvm 01 463 SunOS 5.8: usr platform links
110834 SUNWkvm 03 317 Obsoleted by: 109873-11 SunOS 5.8: SUNW,Sun-Fire-15000 libprtdiag_
110843 SUNWkvm 03 437 Obsoleted by: 110849-06 SunOS 5.8: libprtdiag_psr.so.1 patch for S
110849 SUNWkvm 10 197 SunOS 5.8: PICL support for SUNW,Sun-Fire-880
110853 SUNWkvm 01 441 SunOS 5.8: SUNW,Sun-Fire-880 usr platform links patch
110932 SUNWcsr 01 554 Obsoleted by: 109906-06 SunOS 5.8: /sbin/dhcpagent patch
111096 SUNWcsr 04 135 SunOS 5.8: fcip driver patch
SUNWcsu
111323 SUNWxcu4 01 484 SunOS 5.8: /usr/xpg4/bin/more patch
111393 SUNWatfsu 02 423 SunOS 5.8: /usr/lib/autofs/automountd patch
111406 SUNWcsu 02 185 Netra ct 1.0: Dual Console TTYmux support
111412 SUNWcsu 09 105 SunOS 5.8: Sun StorEdge Traffic Manager patch
111413 SUNWluxop 08 105 SunOS 5.8: luxadm, liba5k and libg_fc patch
111431 SUNWcsl 01 423 Obsoleted by: 108993-07 SunOS 5.8: /usr/lib/libldap.so.4 patch
SUNWcsr
111433 SUNWcar 02 458 Obsoleted by: <Integration> SunOS 5.8: Supplemental Kernel Update
SUNWcsr
111439 SUNWcsr 01 442 SunOS 5.8: /kernel/fs/tmpfs patch
111459 SUNWcar 01 489 Obsoleted by: <INTEGATION> SunOS 5.8: Supplemental kernel update
SUNWcsr
111562 SUNWcsl 02 167 SunOS 5.8: /usr/lib/librt.so.1 patch
111588 SUNWcsr 03 170 SunOS 5.8: /kernel/drv/ws and /kernel/fs/specfs patch
111741 SUNWxwmod 02 274 X11 6.4.1: hwc patch
111791 SUNWkvm 01 316 SunOS 5.8: usr platform links patch for SUNW,Sun-Fire-480R
111793 SUNWkvm 03 202 SunOS 5.8: libprtdiag_psr.so.1 patch for SUNW,Sun-Fire-480R
111796 SUNWcsr 04 157 SunOS 5.8: Remote Shared Memory patch
SUNWcsu
111802 SUNWcsu 01 317 SunOS 5.8: /usr/lib/rcm/modules/SUNW_cluster_rcm.so patch
111804 SUNWcsu 02 311 SunOS 5.8: /usr/sbin/rem_drv patch
111808 SUNWcsu 01 295 SunOS 5.8: /usr/lib/adb/devinfo patch
111823 SUNWfris 01 195 SunOS 5.8: New features UR6 European Support
111831 SUNWcsu 01 386 SunOS 5.8: /usr/kernel/drv/dump patch
111848 SUNWcar 01 401 Obsoleted by: <Integration> SunOS 5.8: Supplemental Kernel Update
SUNWcsr
111850 SUNWcar 02 353 Obsoleted by: <Integration> SunOS 5.8: Supplemental Kernel Update
SUNWcsr
111989 SUNWcsu 01 356 SunOS 5.8: usr/bin/egrep patch
112003 SUNWi15cs 03 216 SunOS 5.8: Unable to load fontset in 64-bit Solaris 8 iso-1 or iso
SUNWi1cs
112050 SUNWesu 01 328 SunOS 5.8: ptree patch
112135 SUNWcsl 01 282 SunOS 5.8:: usr/lib/libmapmalloc.so.1 patch
112160 SUNWkvm 01 205 SunOS 5.8: platform links SUNW,Netra-T12 SUNW,Netra-T4
112161 SUNWkvm 02 204 SunOS 5.8: remove libprtdiag_psr.so.1 of SUNW,Netra-T12 SUNW,Netra
112162 SUNWcsr 03 204 SunOS 5.8: patch Netra T12 Lw8 driver
112163 SUNWcsr 01 204 SunOS 5.8: patch Netra T4 Lombus
112167 SUNWkvm 01 195 SunOS 5.8: patch usr/platform/SUNW,UltraAX-i2 symlink
112171 SUNWcsu 01 204 SunOS 5.8: patch usr/sbin/locator
112220 SUNWcsr 01 282 SunOS 5.8:: kernel/misc/nfssrv patch
112328 SUNWcsu 01 226 SunOS 5.8:: /usr/sbin/rpcbind patch
112345 SUNWcsu 01 230 SunOS 5.8:: /usr/bin/pax patch
112359 SUNWi15cs 01 226 Obsoleted by: 112003-03 SunOS 5.8: 64-bit apps can't create fontse
112369 SUNWcar 01 202 SunOS 5.8: environ driver patch
112371 SUNWcsu 01 202 SunOS 5.8: /usr/bin/ruptime patch

112394 SUNWi15cs 01 188 SunOS 5.8: Print euro and other ext. chars
112607 SUNWcsu 02 156 SunOS 5.8: /usr/bin/on patch
112609 SUNWcsr 01 156 SunOS 5.8: /kernel/drv/le and /kernel/drv/sparcv9/le patch
112670 SUNWcsu 01 118 SunOS 5.8: /usr/sbin/clinfo patch
112852 SUNWcsr 01 111 SunOS 5.8: Supplemental Kernel Update Patch for 108528-14
==========

Appendix B: disable_unneeded_services_at_boot.sh

```
#!/bin/ksh -xv
#####################################################################
#
# Name    : disable_unneeded_services_at_boot.sh
# Author  : John Worthing
# Date    : 5/17/2001
#
# This shell script disables unneeded services at boot by renaming
# various startup scripts in the /etc/rc* directories.  It's meant
# to be run after a Core System Support installation.
#
#####################################################################

DIRS="
/etc/rc0.d
/etc/rc1.d
/etc/rc3.d
"

# First, remove files for run states other than run-level 2

for i in $DIRS
 do
  cd $i
   for j in `ls K* S*`
    do
     mv $j notused.$j
    done
 done

# Next, disable unneeded services in /etc/rc2.d

cd /etc/rc2.d
mv K28nfs.server notused.K28nfs.server
mv S30sysid.net notused.S30sysid.net
mv S71ldap.client notused.S71ldap.client
mv S71rpc notused.S71rpc
mv S71sysid.sys notused.S71sysid.sys
mv S72autoinstall notused.S72autoinstall
mv S73cachefs.daemon notused.S73cachefs.daemon
mv S73nfs.client notused.S73nfs.client
mv S74autofs notused.S74autofs
mv S80PRESERVE notused.S80PRESERVE
mv S93cacheos.finish notused.S93cacheos.finish
```

Appendix C: /etc/init.d/nddconfig; /etc/rc2.d/S70nddconfig

```
#!/sbin/sh
#
# This file was created from the original nddconfig file obtained at
# http://www.sun.com/bluprints/tool/.  It has been edited significantly
# to make it more readable, more Solaris 8 specific, and more tailored
# to the role for which the box was built.

#  This option determines the period of time the Address Resolution
#  Protocol (ARP) cache maintains entries. ARP attacks may be effective
#  with the default interval. Shortening the timeout interval should
#  reduce the effectiveness of such an attack.
#
ndd -set /dev/arp arp_cleanup_interval 60000

#  This option determines whether to forward broadcast packets directed
#  to a specific net or subnet, if that net or subnet is directly
#  connected to the machine. If the system is acting as a router, this
#  option can be exploited to generate a great deal of broadcast network
#  traffic. Turning this option off will help prevent broadcast traffic
#  attacks.
#
ndd -set /dev/ip ip_forward_directed_broadcasts 0

#  This option determines whether to forward packets that are source
#  routed. These packets define the path the packet should take instead
#  of allowing network routers to define the path.
#
ndd -set /dev/ip ip6_forward_src_routed 0

#  This option determines the period of time at which a specific route
#  will be kept, even if currently in use. ARP attacks may be effective
#  with the default interval. Shortening the time interval may reduce
#  the effectiveness of attacks.
#
ndd -set /dev/ip ip_ire_arp_interval 60000

#  This option determines whether to respond to ICMP broadcast echo
#  requests (ping). An attacker may try to create a denial of service
#  attack on subnets by sending many broadcast echo requests to which all
#  systems will respond. This also provides information on systems that
#  are available on the network.
#
ndd -set /dev/ip ip6_respond_to_echo_multicast 0

#  This option determines whether to respond to ICMP timestamp requests
#  which some systems use to discover the time on a remote system. An
#  attacker may use the time information to schedule an attack at a
#  period of time when the system may run a cron job (or other time-
#  based event) or otherwise be busy. It may also be possible predict
#  ID or sequence numbers that are based on the time of day for spoofing
#  services.
#
ndd -set /dev/ip ip_respond_to_timestamp 0
```

```
#  This option determines whether to respond to ICMP broadcast timestamp
#  requests which are used to discover the time on all systems in the
#  broadcast range. This option is dangerous for the same reasons as
#  responding to a single timestamp request. Additionally, an attacker
#  may try to create a denial of service attack by generating many
#  broadcast timestamp requests.
#
ndd -set /dev/ip ip_respond_to_timestamp_broadcast 0

#  This option determines whether to enable strict destination
#  multihoming. If this is set to 1 and ip_forwarding is set to 0, then
#  a packet sent to an interface from which it did not arrive will be
#  dropped. This setting prevents an attacker from passing packets across
#  a machine with multiple interfaces that is not acting a router.
#
ndd -set /dev/ip ip6_strict_dst_multihoming 1
```

Appendix D: sample /etc/syslog.conf

```
#ident  "@(#)syslog.conf        1.5     98/12/14 SMI"   /* SunOS 5.0 */
#
# Copyright (c) 1991-1998 by Sun Microsystems, Inc.
# All rights reserved.
#
# syslog configuration file.
#
# ******** TABS, no spaces in this file. ********
#
# This file is processed by m4 so be careful to quote (`') names
# that match m4 reserved words.  Also, within ifdef's, arguments
# containing commas must be quoted.
#
*.err;kern.notice;auth.notice           /dev/sysmsg
*.err;kern.debug;daemon.notice;mail.crit        /var/adm/messages

*.alert;kern.err;daemon.err             operator
*.alert                         root

*.emerg                         *

# Additional logging not included in the default selectors above.
#
auth.notice                     /var/log/authlog
mail.debug                      /var/log/syslog
*.info;mail.none                /var/adm/messages
local2.debug                    /var/log/sudo.log

# non-loghost machines will use the following lines to cause "user"
# log messages to be logged locally.
#
ifdef(`LOGHOST', ,
user.err                        /dev/sysmsg
user.err                        /var/adm/messages
user.alert                      `root, operator'
user.emerg                      *
)

# Centralized logging of sudo as well as most kernel and mail messages.
# The *.none facility levels help to reduce duplicate entries in remote logs.
#
local2.debug                                    @logmaster      # sudo on local2.debug
*.info;kern.none;mail.none;local2.none          @logmaster      # gets *.info without duplications
mail.debug                                      @logmaster      # gets the sendmail stuff
*.notice;kern.debug;*.err                       @logmaster      # this one gets the juicy leftovers
```

Appendix E: sample swatch configuration file from central log server (@logmaster)

```
# .swatchrc - Swatch configuration file for constant monitoring
#

# Bad login attempts
watchfor      /INVALID|REPEATED|INCOMPLETE/
                    mail=jworthing@jii.com

# System crashes and halts
watchfor      /(panic|halt)/
          mail=jworthing@jii.com

# System reboots
watchfor      /SunOS Release/
          mail=jworthing@jii.com

# Unplugged or bad ethernet cable
watchfor      /Link Down/
          mail=jworthing@jii.com

# Ignore commands issued by me since I know I did them
watchfor      /jworthin/
          ignore

# SU attempts
watchfor      /su: .* failed|su: .* succeeded/
          mail=jworthing@jii.com
                    throttle 01:00

# Changing of sudo configuration
watchfor      /sudo.log|visudo|sudoers/
          mail=jworthing@jii.com
          throttle 01:00

# Changing of syslog configuration or log file
watchfor      /messages|syslog|namedlog/
          mail=jworthing@jii.com
          throttle 01:00

# Changing of swatch configuration
watchfor      /swatch|.swatchrc/
                    mail=jworthing@jii.com
                    throttle 01:00

# Changing of openssh configuration
watchfor      /sshd_config|ssh_config|sshd/
                    mail=jworthing@jii.com
                    throttle 01:00

# Changing of passwd or shadow file
watchfor      /passwd|shadow|opasswd|oshadow/
                    mail=jworthing@jii.com
                    throttle 01:00
```

```
# Changing telnet or ftp access
watchfor     /inetd|ftpusers|login/
                mail=jworthing@jii.com
                throttle 01:00

# Watch for important OS events
watchfor     /config*|make|patchadd|pkgadd|pkgrm|patchrm|install*|format/
                mail=jworthing@jii.com
                throttle 01:00

# Watch for important user events
watchfor     /shutdown|init|reboot|uadmin|chmod|chown|su /
                mail=jworthing@jii.com
                throttle 01:00

# Watch for important application events
watchfor     /rndc|named|sendmail|mqueue/
                mail=jworthing@jii.com
                throttle 01:00
```

Appendix F: Updated /usr/lib/newsyslog

```
#! /bin/sh
#
# Copyright(c) 1997, by Sun Microsystems, Inc.
# All rights reserved.
#
#ident @Z%newsyslog  1.3  97/03/31 SMI
#
LOG=messages
cd /var/adm
test -f $LOG.29 && mv $LOG.5 $LOG.30
test -f $LOG.28 && mv $LOG.4 $LOG.29
test -f $LOG.27 && mv $LOG.3 $LOG.28
test -f $LOG.26 && mv $LOG.2 $LOG.27
test -f $LOG.25 && mv $LOG.1 $LOG.26
test -f $LOG.24 && mv $LOG.0 $LOG.25
test -f $LOG.23 && mv $LOG.7 $LOG.24
test -f $LOG.22 && mv $LOG.6 $LOG.23
test -f $LOG.21 && mv $LOG.5 $LOG.22
test -f $LOG.20 && mv $LOG.4 $LOG.21
test -f $LOG.19 && mv $LOG.3 $LOG.20
test -f $LOG.18 && mv $LOG.2 $LOG.19
test -f $LOG.17 && mv $LOG.1 $LOG.18
test -f $LOG.16 && mv $LOG.0 $LOG.17
test -f $LOG.15 && mv $LOG.7 $LOG.16
test -f $LOG.14 && mv $LOG.6 $LOG.15
test -f $LOG.13 && mv $LOG.5 $LOG.14
test -f $LOG.12 && mv $LOG.4 $LOG.13
test -f $LOG.11 && mv $LOG.3 $LOG.12
test -f $LOG.10 && mv $LOG.2 $LOG.11
test -f $LOG.9 && mv $LOG.1 $LOG.10
test -f $LOG.8 && mv $LOG.0 $LOG.9
test -f $LOG.7 && mv $LOG.7 $LOG.8
test -f $LOG.6 && mv $LOG.6 $LOG.7
test -f $LOG.5 && mv $LOG.5 $LOG.6
test -f $LOG.4 && mv $LOG.4 $LOG.5
test -f $LOG.3 && mv $LOG.3 $LOG.4
test -f $LOG.2 && mv $LOG.2 $LOG.3
test -f $LOG.1 && mv $LOG.1 $LOG.2
test -f $LOG.0 && mv $LOG.0 $LOG.1
mv $LOG   $LOG.0
cp /dev/null $LOG
chmod 600    $LOG
#
LOGDIR=/var/log
LOG=syslog
if test -d $LOGDIR
then
     cd $LOGDIR
     if test -s $LOG
     then
                    test -f $LOG.29 && mv $LOG.5 $LOG.30
                    test -f $LOG.28 && mv $LOG.4 $LOG.29
                    test -f $LOG.27 && mv $LOG.3 $LOG.28
                    test -f $LOG.26 && mv $LOG.2 $LOG.27
```

```
                test -f $LOG.25 && mv $LOG.1 $LOG.26
                test -f $LOG.24 && mv $LOG.0 $LOG.25
                test -f $LOG.23 && mv $LOG.7 $LOG.24
                test -f $LOG.22 && mv $LOG.6 $LOG.23
                test -f $LOG.21 && mv $LOG.5 $LOG.22
                test -f $LOG.20 && mv $LOG.4 $LOG.21
                test -f $LOG.19 && mv $LOG.3 $LOG.20
                test -f $LOG.18 && mv $LOG.2 $LOG.19
                test -f $LOG.17 && mv $LOG.1 $LOG.18
                test -f $LOG.16 && mv $LOG.0 $LOG.17
                test -f $LOG.15 && mv $LOG.7 $LOG.16
                test -f $LOG.14 && mv $LOG.6 $LOG.15
                test -f $LOG.13 && mv $LOG.5 $LOG.14
                test -f $LOG.12 && mv $LOG.4 $LOG.13
                test -f $LOG.11 && mv $LOG.3 $LOG.12
                test -f $LOG.10 && mv $LOG.2 $LOG.11
                test -f $LOG.9 && mv $LOG.1 $LOG.10
                test -f $LOG.8 && mv $LOG.0 $LOG.9
        test -f $LOG.7 && mv $LOG.7  $LOG.8
        test -f $LOG.6 && mv $LOG.6  $LOG.7
        test -f $LOG.5 && mv $LOG.5  $LOG.6
        test -f $LOG.4 && mv $LOG.4  $LOG.5
        test -f $LOG.3 && mv $LOG.3  $LOG.4
        test -f $LOG.2 && mv $LOG.2  $LOG.3
        test -f $LOG.1 && mv $LOG.1  $LOG.2
        test -f $LOG.0 && mv $LOG.0  $LOG.1
        mv $LOG    $LOG.0
        cp /dev/null $LOG
        chmod 600    $LOG
        sleep 40
    fi
fi

kill -HUP `cat /etc/syslog.pid`
```

Appendix G: Sample /etc/sudoers file

```
# Sudo has two main objectives on this server: protect the root password and
# provide an audit trail for those who use it. Sudo is not allowed to run the su
# binaries and swatch is watching the logs for anyone who might tamper with
# su (for example, someone copying it to a different location). This is all in an
# effort to ensure that only the users that actually know the root password can
# run su. The wheel group furthers this end as well. And the fix-modes script
# helps to change most binaries to root ownership, forcing the use of sudo to
# accomplish most tasks. Hence, most activity on the server will be logged via
# sudo.

# Host alias specification

# User alias specification

User_Alias  ADMINS = jworthin,admin1,admin2

# Cmnd alias specification

Cmnd_AliasROOT_ONLY = /usr/bin/su, /sbin/su, /platform/sun4u/kernel/drv/su

# Defaults specification

# User privilege specification
rootALL=(ALL) ALL
ADMINS      ALL=(ALL) ALL, !ROOT_ONLY

# Uncomment to allow people in group wheel to run all commands
# %wheel    ALL=(ALL)  ALL

# Same thing without a password
# %wheel    ALL=(ALL)  NOPASSWD: ALL

# Samples
# %users  ALL=/sbin/mount /cdrom,/sbin/umount /cdrom
# %users  localhost=/sbin/shutdown -h now
```

Appendix H: /usr/local/etc/ssh_config

```
#    $OpenBSD: ssh_config,v 1.15 2002/06/20 20:03:34 stevesk Exp $

# This is the ssh client system-wide configuration file.  See
# ssh_config(5) for more information.  This file provides defaults for
# users, and the values can be changed in per-user configuration files
# or on the command line.

# Configuration data is parsed as follows:
#  1. command line options
#  2. user-specific file
#  3. system-wide file
# Any configuration value is only changed the first time it is set.
# Thus, host-specific definitions should be at the beginning of the
# configuration file, and defaults at the end.

# Site-wide defaults for various options

# Host *
#   ForwardAgent no
#   ForwardX11 no
#   RhostsAuthentication no
#   RhostsRSAAuthentication no
#   RSAAuthentication yes
#   PasswordAuthentication yes
#   BatchMode no
#   CheckHostIP yes
#   StrictHostKeyChecking ask
#   IdentityFile ~/.ssh/identity
#   IdentityFile ~/.ssh/id_rsa
#   IdentityFile ~/.ssh/id_dsa
#   Port 22
#   Protocol 2,1
#   Cipher 3des
#   Ciphers aes128-cbc,3des-cbc,blowfish-cbc,cast128-cbc,arcfour,aes192-cbc,aes256-cbc
#   EscapeChar ~
```

Appendix I: /usr/local/etc/sshd_config

```
#	$OpenBSD: sshd_config,v 1.56 2002/06/20 23:37:12 markus Exp $

# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/bin:/bin:/usr/sbin:/sbin:/usr/local/bin

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options change a
# default value.

Port 22
#Protocol 2,1
#ListenAddress 0.0.0.0
#ListenAddress ::

# HostKey for protocol version 1
HostKey /usr/local/etc/ssh_host_key
# HostKeys for protocol version 2
HostKey /usr/local/etc/ssh_host_rsa_key
HostKey /usr/local/etc/ssh_host_dsa_key

# Lifetime and size of ephemeral version 1 server key
KeyRegenerationInterval 3600
ServerKeyBits 768

# Logging
#obsoletes QuietMode and FascistLogging
SyslogFacility AUTH
LogLevel INFO

# Authentication:

LoginGraceTime 600
PermitRootLogin no
StrictModes yes

# Added security:
# Only allow authorized users and groups access to sshd, and
# always display a banner warding off unauthorized users.

AllowUsers jworthin admin2 admin3
AllowGroups sysadmin wheel
Banner /etc/issue

RSAAuthentication yes
PubkeyAuthentication yes
AuthorizedKeysFile .ssh/authorized_keys

# rhosts authentication should not be used
RhostsAuthentication no
# Don't read the user's ~/.rhosts and ~/.shosts files
IgnoreRhosts yes
```

```
# For this to work you will also need host keys in /usr/local/etc/ssh_known_hosts
RhostsRSAAuthentication no
# similar for protocol version 2
HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# RhostsRSAAuthentication and HostbasedAuthentication
IgnoreUserKnownHosts no

# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication yes
PermitEmptyPasswords no

# Change to no to disable s/key passwords
ChallengeResponseAuthentication yes

# Kerberos options
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes

#AFSTokenPassing no

# Kerberos TGT Passing only works with the AFS kaserver
#KerberosTgtPassing no

# Set this to 'yes' to enable PAM keyboard-interactive authentication
# Warning: enabling this may bypass the setting of 'PasswordAuthentication'
#PAMAuthenticationViaKbdInt yes

X11Forwarding no
X11DisplayOffset 10
X11UseLocalhost yes
PrintMotd yes
#PrintLastLog yes
KeepAlive yes
#UseLogin no
UsePrivilegeSeparation yes
Compression yes

#MaxStartups 10
# no default banner path
#Banner /some/path
#VerifyReverseMapping no

# override default of no subsystems
Subsystem  sftp /usr/local/libexec/sftp-server
```

Appendix J: /etc/init.d/sshd; /etc/rc2.d/S86sshd

```
#!/sbin/sh
#
# sshd startup script.
#

case "$1" in
'start')
        if [ -f /usr/local/sbin/sshd ]; then
                /bin/echo 'Secure shell service starting.'
                /usr/local/sbin/sshd >/dev/console 2>&1 &
        fi
        ;;

'stop')
        if [ -f /var/run/sshd.pid ]; then
                syspid=`/bin/cat /var/run/sshd.pid`
                [ "$syspid" -gt 0 ] && /bin/kill -TERM $syspid
        fi
        ;;

*)
        /bin/echo "Usage: $0 { start | stop }"
        exit 1
        ;;
esac
```

Appendix K: /etc/named.conf

```
// Configuration file for DNS services on JII.com
// Modified: September 24, 2002

// This key was generated using rndc-confgen.

key "rndc-key" {
     algorithm hmac-md5;
     secret "fFln+QxfJkqai7l/3WDFAg==";
};

controls {
     inet 127.0.0.1 port 953
          allow { 127.0.0.1; } keys { "rndc-key"; };
};

// Define access lists. The internal_nets group contains systems that are only on the local
// network and should be first in the list of ACLs. The allowed_nets is both the internal_nets
// group and other systems that are allowed to use this DNS server. The value of ACLs from
// a security point of view are obvious - similar to tcp wrappers, ACLs help to ensure that the
// more legitimate users, and less unauthorized users, have access to these DNS services.

acl internal_nets {
   127.0.0.1;
   10.1.1/24;
   192.168/16;
};

acl allowed_nets {
   127.0.0.1;
   10.1.1/24;
   192.168/16;
};

// This access list contains specific IP addresses that we do not want speaking to us from the
// outside world, because they are more than likely spoofed. The addresses listed here are
// private networks and multicasted network spaces. 10.1.1.x has been excluded for our internal
// network. The most recent list of these addresses can be found at www.iana.org.

acl denied_nets {
   ! 10.1.1/24; 10/8;
   ! 192.168/16; 192.168/16;
   ! 127.0.0.1; 127/8;
   172.16/12;
   224/8;
   225/8;
   226/8;
   227/8;
   228/8;
   229/8;
   230/8;
   231/8;
   232/8;
   233/8;
   234/8;
```

```
    235/8;
    236/8;
    237/8;
    238/8;
    239/8;
};

// Main server configuration. This section sets the server defaults. The default settings here can
// be over-ridden by putting the option in the zone entry.

options {
    version "JII DNS server";
    directory "/var/named";
    pid-file "/var/run/named.pid";
    statistics-file "/var/named/namedlog.stats";
    dump-file "/var/named/namedlog.dump";
    random-device "/dev/random";
    zone-statistics yes;

    notify no;                    // prevent DOS attacks
    listen-on-v6 { none; };       // do not listen for ip-v6
    transfer-format many-answers; // more efficient zone transfers
    max-transfer-time-in 60;      // maximum zone transfer time
    interface-interval 0;         // disable dynamic interfaces

    allow-transfer { none; };     // transfers configured in zones
    allow-query { allowed_nets; };
    blackhole { denied_nets; };
};

// Logging configuration. Define what will be logged and keep in a separate file for easy
// maintenance. Instead of commenting out or deleting a category, assign it to "null" for later
// use. From a security point of view, every application that runs on a server should write to a
// log for debugging and review. BIND is certainly no different.

logging {
    channel "defaultlog" {
        file "/var/adm/logs/namedlog";
        severity info;
        print-category yes;
        print-severity yes;
        print-time yes;
    };
    channel "statslog" {
        file "/var/adm/logs/namedstats";
        severity info;
    };
    category default { defaultlog; };
    category general { defaultlog; };
    category database { defaultlog; };
    category security { defaultlog; };
    category config { defaultlog; };
    category resolver { defaultlog; };
    category xfer-in { defaultlog; };
    category xfer-out { defaultlog; };
    category notify { defaultlog; };
```

```
    category client { defaultlog; };
    category unmatched { defaultlog; };
    category network { defaultlog; };
    category update { defaultlog; };
    category queries { null; };
    category dispatch { defaultlog; };
    category dnssec { defaultlog; };
    category lame-servers { null; };
};

//  Define views for this server.  Be sure to include the internal net first so that it doesn't get
//  blocked by rules in subsequent views.  The views in BIND9 provide a very needed function
//  for organizations running large , over-crowded domains.  Views enable you to create an
//  internal and external view of your domain.  This is significant because you can create an
//  internal view such that only the non-routable, private IPs can be queried.  And then you can
//  do the opposite for the external view, only displaying external, routable hosts and IPs.

server 192.168.127.20 {       //  tell the slave to use the same TSIG key to sign all requests to
     keys { "rndc-keys"; };   //  the master when requesting a zone transfer.
};

view "internal-zone" in {
    match-clients { internal_nets; };
    additional-from-auth yes;
    additional-from-cache yes;
    recursion yes;

    zone "." in {
        type hint;
        file "named.cache";
    };

    zone "0.0.127.in-addr.arpa" in {
        type master;
        file "internal/db.127.0.0";
        allow-query { any; };
    };

    zone "1.1.10.in-addr.arpa" in {
        type slave;
       masters{ 192.168.127.20 key "rndc-keys"; };
        file "internal/db.10.1.1";
    };

    zone "1.168.192.in-addr.arpa" in {
        type slave;
       masters{ 192.168.127.20 key "rndc-keys"; };
        file "internal/db.192.168.1";
    };

    zone "jii.com" in {
        type slave;
       masters{ 192.168.127.20 key "rndc-keys"; };
        file "internal/db.jii.com";
    };
```

```
};

view "external-zone" in {
    match-clients { any; };
    additional-from-auth no;
    additional-from-cache no;
    recursion no;

    zone "." in {
        type hint;
        file "named.cache";
    };

    zone "jii.com" in {
        type slave;
       masters{ 192.168.127.20 key "rndc-keys"; };
        file "external/db.jii.com";
        allow-query { any; };
    };

};

// Define a view for clients perusing the CHAOS class. This is mainly from a support point of view.

view "external-chaos" chaos {
    match-clients { any; };
    recursion no;

    zone "." {
        type hint;
        file "/dev/null";
    };

    zone "bind" {
        type slave;
        file "db.bind";
       masters{ 192.168.127.20 key "rndc-keys"; };
        allow-query { internal_nets; };
        allow-transfer { none; };
    };
};

//
//  end of configuration file
```

Appendix L: /etc/init.d/named; /etc/rc2.d/S50named

```
#!/sbin/sh
# Paths to key files:
named="/opt/bind/sbin/named";   # relative to $root
uid="named";
root='/chroot';
pid="$root/var/run/named.pid";

case $1 in
'start')
        # make sure named is not running
        [ -f $pid ] && /bin/kill `/bin/cat $pid` >/dev/null 2>&1

        /bin/echo "Starting DNS services `/bin/date`. . . \c"
        if [-f $named -u $uid -t $root ]; then
                    $named -u $uid -t $root &
          /bin/sleep 1
          if [ "$?" -ne 0 ]; then
            /bin/echo "Warning: named did not start"
          elif [ ! -f $pid ]; then
            /bin/echo "Warning: named pid file $pid missing."
          else
            /bin/echo "named running with pid `/bin/cat $pid`".
          fi
        fi
        ;;
'stop')
        /bin/echo "Stopping DNS services `/bin/date`. . .\c"
        /bin/kill `/bin/cat $pid`
        if [ "$?" -ne 0 ]; then
          /bin/echo "Warning: named not killed"
        else
          /bin/echo "done."
        fi
        ;;
'restart')
        /bin/echo "Restarting DNS services `/bin/date`. . .\c"
        $0 stop
        /bin/sleep 1
        $0 start
        ;;
'reload')
        /bin/echo "Reloading DNS services `/bin/date`. . .\c"
        /bin/kill -1 `/bin/cat $pid`
        if [ "$?" -ne 0 ]; then
          /bin/echo "Warning: named not reloaded"
        else
          /bin/echo "HUP sent to PID `/bin/cat $pid`".
        fi
        ;;
*)
        /bin/echo "Usage: $0 { start | stop | restart | reload}"
        ;;
esac
exit 0
```

Appendix M: config.h for Tripwire ASR

```
/* $Id: config.h,v 1.5 1994/07/15 11:02:52 gkim Exp $ */

/*
 * config.h
 *
 *    Tripwire configuration file
 *
 * Gene Kim
 * Purdue University
 */


/***
 *** Operating System specifics
 ***
 *** Look in the ./configs directory, and include appropriate header
 *** file that corresponds with your operating system.
 ***/

#include "../configs/conf-svr4.h"

#ifdef TW_TYPE32
typedef TW_TYPE32 int32;
typedef unsigned TW_TYPE32 uint32;
#else
typedef long int32;
typedef unsigned long uint32;
#endif

/***
 *** SYSTEM SPECIFIC Tripwire Configuration
 ***/

/******* signature functions ****************************************
 *
 * Choose among these:
 *
 *    sig_md5_get          : MD5 function
 *                    (the RSA Data Security, Inc. MD5 Message-
 *                     Digesting Algorithm)
 *    sig_snefru_get       : Snefru function
 *                    (the Xerox Secure Hash Function)
 *    sig_null_get     : null function (returns 0 for all)
 *
 *  By default, Tripwire uses
 *          int (pf_signature0)() = sig_null_get;
 *          int (pf_signature1)() = sig_md5_get;
 *          int (pf_signature2)() = sig_snefru_get;
 *
 *  However, since Snefru is comparatively computationally expensive, you
 *  might consider using only MD5.  This can be done in the configuration,
 *  however, and should not be done by defining away the signature here.
 *
 *  You can replace one of the signature algorithms with another of your
```

```
 *  own choice by adding it to the build procedure, and putting it in
 *  here in place of one of these standard routines.  See the design
 *  document for hints on this.
 *
 *  To do this, just set one of the signature function pointers to
 *  your own function.
 *
 ************************************************************************/

#define SIG0FUNC sig_null_get
#define SIG1FUNC sig_md5_get
#define SIG2FUNC sig_snefru_get
#define SIG3FUNC sig_crc32_get
#define SIG4FUNC sig_crc_get
#define SIG5FUNC sig_md4_get
#define SIG6FUNC sig_md2_get
#define SIG7FUNC sig_sha_get
#define SIG8FUNC sig_haval_get
#define SIG9FUNC sig_null_get

#define SIG0NAME "nullsig"
#define SIG1NAME "md5"
#define SIG2NAME "snefru"
#define SIG3NAME "crc32"
#define SIG4NAME "crc16"
#define SIG5NAME "md4"
#define SIG6NAME "md2"
#define SIG7NAME "sha"
#define SIG8NAME "haval"
#define SIG9NAME "nullsig"

/******* path to Tripwire files ***********************************
 *
 *    Ideally, CONFIG_PATH and DATABASE_PATH should be pointing to
 *    some read-only media, or some filesystem mounted remotely
 *    from a "secure-server".  (See design document for details.)
 *
 *    Note:  No trailing '/' in the paths!
 *
 ************************************************************************/

#if !defined(SYSV) || (defined(SYSV) && (SYSV > 2))
 #define CONFIG_PATH     "/opt/tw/adm/tcheck"
 #define DATABASE_PATH   "/opt/tw/adm/tcheck/databases"
#else
 #define CONFIG_PATH     "/opt/tw/adm/tcheck"
 #define DATABASE_PATH   "/opt/tw/adm/tcheck/databases"
#endif

#define CONFIG_PATH     "/opt/tw/bin/tw"
#define DATABASE_PATH   "/opt/tw/var/tripwire"

/******* name of Tripwire files ***********************************
 *
 *    Static filenames are nice, but we allow run-time binding to
 *    support multiple hosts sharing the same directory (without
```

```
 *    having to recompile.
 *
 *    Use the '@' character to represent the hostname of the machine
 *    running Tripwire.
 *
 *    For example "tw.db_@" would expand to:
 *
 *         tw.db_mentor.cc.purdue.edu
 *
 ************************************************************************/

#define CONFIG_FILE  "tw.config"
#define DATABASE_FILE    "tw.db_@"

/******* Default ignore mask ***************************************
 *
 *    Usually, the only thing you want to ignore is the access time
 *    stamp.  But there may be applications where you want to know
 *    about any accesses, too.
 *
 *    Similarly, there may be some environments where you can have a much
 *    more forgiving ignore mask.
 *
 *    By default, Tripwire uses:
 *         "R" --  read-only files, where only the access time
 *              stamp can change.
 *      Alternatively, you might want to make the default be "R-2"
 *      This would be faster than simply "R", at some small loss
 *    (perhaps) of protection.
 *
 *    NOTE:  Users with backup programs that read through the file
 *    system rather than the raw disk (e.g., bru and cpio) should
 *    add a "-c" to the DEFAULTIGNORE string.  Otherwise, every file
 *    will be reported as changed after backups.
 *
 ************************************************************************/

#define DEFAULTIGNORE  "R-23456789"

/******* Temporary file template ***********************************
 *
 *    Usually, temporary files are stored in /tmp.  You may want
 *    to use a different directory if your system does not support
 *    the BSD "sticky" bit on directories.  (i.e., only owner or root
 *    can rename or delete files.)
 *
 *    Make sure that there are at least 6 X's in the template.
 *    Each consecutive X signifies a number that mktemp() can
 *    fill in with a random number.
 *
 ************************************************************************/

#define TEMPFILE_TEMPLATE "/tmp/twzXXXXXX"
```

Appendix N: Makefile for Tripwire 1.3.1-2 ASR

```
#  Tripwire build
#

###
###   Start of user-modified settings
###   Examine these and change the ones that need to be
###   Altered on your system
###

# destination directory for final executables
DESTDIR = /opt/tw/bin/tw
DATADIR = /opt/tw/var/tripwire

# destination for man pages
MANDIR  = /usr/man          # This needs to change to reflect the path
                  # on your system

# system utilities
LEX     = lex
#LEX    = flex              # For the GNU crowd

YACC   = yacc
#YACC = bison -y         # For the GNU crowd (make it look like yacc)
#                  # see ./contrib/README.linux for tips on
#                  # making work.

# for SVR4 make (must be a Bourne-type shell)
SHELL  = /bin/sh
#SHELL      = /bin/ksh          # Another common shell
#SHELL      = /bin/bash         # For the GNU fanatics

# you can use ANSI C if you like, but K&R is equally fine.
#CC     = cc            # common
CC     = gcc           # also common
#CC     = /usr/ccs/bin/cc# Pyramid DC/OSx (SVR4)

CFLAGS = -O              # common
#CFLAGS = -g             # common
#CFLAGS = -g             # debugging
#CFLAGS = -O -cckr       # SGI
#    NOTE: some versions of the HP C compiler optimizer breaks snefru.c!
#          consider recompiling this file seperately without optimization
#CFLAGS = -O -Aa -N      # HP/UX ansi
#CFLAGS = -O -Ac -N      # HP/UX K&R
#CFLAGS = -O -Ac -N -Wl,-a,archive    # HP/UX K&R, insure archived, static link
#CFLAGS = -systype bsd43 # ETA/10 (SVR3)
#CFLAGS = -systype bsd43 # MIPS RISC/OS 4.5x
#CFLAGS = -O -ansi       # gnu CC
#CFLAGS = -O -ansi -W -Wreturn-type -Wswitch -Wshadow # gnu CC w/all warnings
#CFLAGS = -OG            # Pyramid OSx
#CFLAGS = -O -Kold       # Pyramid DC/OSx (SVR4)
#CFLAGS = -DTW_TYPE32='int'     # DEC OSF/1 Alpha (or any other architecture
                  # where int [but not long] is a 32 bit quantity)
```

```
# a C preprocessor (to build inode.h)
CPP     = $(CC) -E              # common
#CPP    = /usr/lib/cpp          # on older systems
#CPP    = /lib/cpp          # on older systems

# make sure libraries are not linked dynamically (as a security measure)
#LDFLAGS= -static       # Most systems, Linux / RedHat 5.2 and previous
LDFLAGS= -ldl         # Solaris 2.x, Redhat 6.0
# common
#LDFLAGS= -non_shared         # OSF/1
#LDFLAGS= -Bstatic          # SunOS 4 (cannot statically link tripwire
                        #         on Solaris 2.3)
#LDFLAGS= -dn                # Pyramid DC/OSx (SVR4)

# libraries
LIBS =                  # common
#LIBS = -lsocket              # SCO
#LIBS = -lmalloc -lsun -lc_s # IRIX 4.0
#LIBS = -lmalloc -lsun              # IRIX 6.x
#LIBS = -lx              # Xenix
#LIBS = -lbsd              # MIPS RISC/OS
#LIBS = -lgnumalloc          # Encore / UMAX V


# If you don't have the install command, you need to replace
# the use of it later in the makefile with a cp and chmod
INSTALL= /usr/ucb/install    # common
#INSTALL= /usr/ucb/install  # Pyramid DC/OSx (SVR4)
#INSTALL= /etc/install        # Pyramid OSx, IRIX 6.x
#INSTALL= /bin/cp       # no install
#INSTALL= /usr/bin/installbsd    # OSF/1 (DEC only?)

# how you get hostname information (BSD vs. SYSV style)
HOSTNAME     = "hostname"         # BSD
#HOSTNAME   = "uname -n"         # System V


###
###   End of user-modified settings
###   You should not need to change anything after this
###

DIST         = tripwire-1.3.1

all:
     (cd util; make CC=$(CC) CFLAGS="$(CFLAGS)"  \
          LDFLAGS="$(LDFLAGS)" CPP="$(CPP)" SHELL=$(SHELL) all)
     (cd src; make CC=$(CC) CFLAGS="$(CFLAGS)" LIBS="$(LIBS)" \
          LDFLAGS="$(LDFLAGS)" CPP="$(CPP)" SHELL=$(SHELL) \
          YACC="$(YACC)" LEX="$(LEX)" all)

install: all
     $(INSTALL) -d $(DESTDIR)
     (cd src; make INSTALL=$(INSTALL) DESTDIR=$(DESTDIR) install)
     (cd man; make INSTALL=$(INSTALL) MANDIR=$(MANDIR) install)
     (cd configs; $(INSTALL) -m 444 tw.config $(DESTDIR))
```

```
	chmod 555 $(DESTDIR)
	$(INSTALL) -m 0755 -d $(DATADIR)
	$(INSTALL) -m 444 tests/tw.db_TEST $(DATADIR)

test:	all
	(cd tests; make HOSTNAME=$(HOSTNAME) DIST=$(DIST) SHELL=$(SHELL) \
		CC=$(CC))

clean:
	(cd src; make clean)
	(cd man; make clean)
	(cd util; make clean)
	(cd tests; make clean)
	rm -f core

clobber: clean
	(cd src; make clobber)
	(cd man; make clean)
	(cd util; make clean)
	(cd tests; make clean)
	rm -f core
	rm -f */*_pure_*.o sigs/*/*_pure_*.o
	rm -rf databases
```

Appendix O: /opt/tw/bin/tw/tw.config file for Tripwire 1.3.1-2 ASR

```
# TripWire Configuration File
# tripwire.conf -- Sept. 27, 2002
#
# Arguments to be used with tripwire on this system;
#
#  -update [[pathname|entry] ...], -- update [[pathname|entry] ...]
#     Database Update mode.  This mode updates the specified pathname or
#     entry in the database.  If the argument provided is a file, only that
#     file is updated.  If the argument is a directory, that directory and
#     all of its children are updated.  If the argument is an entry in the
#     tw.config file, the entire entry in the database is updated.
#
#  -interactive, --interactive
#     Interactive Update mode.  Tripwire first reports all added, deleted,
#     and changed files, then allows the user to update the entry in the
#     database. Note that Tripwire opens up /dev/tty instead of using stdin.
#     This prevents the automation of interactive updates, reducing the
#     chance of system administrators inadvertently updating entries.
#     Updating the database should always be done with care and
#     deliberation.
#


# Monitor the root directory, be sure to include the (=) so tripwire
# doesn't monitor "all" subdirectories.

=/                          R

# Monitor the UNIX kernel itself.

/kernel/genunix             R

# Monitor configuration files needed by sendmail.  The sendmail
# executable will be monitored below in a directory entry.  The
# hosts file will be monitored by /etc/inet directory.

/etc/mail/aliases           R   # will rarely change
/etc/mail/local-host-names  R
/etc/mail/sendmail.cf       R
/etc/mail/submit.cf         R

# Other files / directories to be monitored in /etc.  These are all
# files that are not linked to another directory.

/etc/.login                 R
/etc/auto_home              R
/etc/auto_master            R
/etc/cron.d                 R   # directory
/etc/coreadm.conf           R
/etc/default                R   # directory
/etc/defaultrouter          R
/etc/device.tab             R
/etc/devlink.tab            R
/etc/dumpadm.conf           R
```

```
/etc/format.dat         R
/etc/group              R  # will rarely change
/etc/hosts              R  # will rarely change
/etc/hostname.hme0      R
/etc/hosts.allow        R  # will rarely change
/etc/hosts.deny         R  # should never
/etc/ftpusers           R  # should never
/etc/inet/inetd.conf    R  # may rarely change
/etc/init.d             R  # directory
/etc/inittab            R
/etc/issue              R
/etc/lib                R  # directory
/etc/magic              R
/etc/motd               R
/etc/named.conf         R
/etc/netconfig          R
/etc/nfssec.conf        R
/etc/nodename           R
/etc/nscd.conf          R
/etc/nsswitch.conf      R
/etc/ntp.conf           R
/etc/opt                R  # directory
/etc/opasswd            R
/etc/oshadow            R
/etc/pam.conf           R
/etc/passwd             R
/etc/profile            R
/etc/publickey          R
/etc/rc0.d              R  # directory
/etc/rc1.d              R  # directory
/etc/rc2.d              R  # directory
/etc/rc3.d              R  # directory
/etc/rcS.d              R  # directory
/etc/remote             R
/etc/resolv.conf        R
/etc/rmmount.conf       R
/etc/rndc.conf          R
/etc/rndc.key           R
/etc/rpc                R
/etc/rpld.conf          R
/etc/security           R  # directory
/etc/services           R
/etc/shadow             R
/etc/sudoers            R
/etc/syslog.conf        R
/etc/system             R
/etc/ttydefs            L
/etc/ttysrch            R
/etc/user_attr          R
/etc/vfstab             R

# Other directories to monitor on this system.

/kernel                 R  # directory
=/export                R  # directory
```

```
/export/sysadmin        R   # directory
=/export/home           R   # directory
/opt                    R   # directory
/bin                    R   # directory
/sbin                   R   # directory
/.ssh                   R   # directory


=/var                   L   # directory
=/var/adm               L   # directory
/var/adm/wtmpx          L
=/var/adm/sa            L   # directory
=/var/spool             L   # directory
/var/yp                 L   # directory
/var/spool/cron         L   # directory

=/usr                   R   # directory
/usr/bin                R   # directory
/usr/lib                R   # directory
/usr/sbin               R   # directory
/usr/local              R   # directory

=/chroot                R   # directory
/chroot/etc             R   # directory
/chroot/dev             L-am   # directory
/chroot/var             L   # directory
!/chroot/var/adm/logs   L   # directory


# Special files/directories to monitor.

/var/tmp                # directory
/mnt
/dev                    L-am   # directory

#
# eof
#
```

Appendix P: /opt/tw/tw.check

```
#!/bin/sh
##############################################################################
#
# Name   : tw.check
# Author : John Worthing
# Date   : 7/1/2001
#
# This shell script automates daily disk integrity checks using tripwire.
# It runs tripwire in integrity checking mode and then sends the resulting
# report via mail.
#
##############################################################################

ADMINS=jworthing@jii.com,admin2@jii.com,admin3@jii.com

/opt/tw/bin/tripwire | (cat <<EOF

This is an automated report of possible file integrity changes, generated by
the Tripwire integrity checker. To tell Tripwire that a file or entire
directory tree is valid, as root run:

/opt/tw/bin/tripwire -update [pathname|entry]

If you wish to enter an interactive integrity checking and verification
session, as root run:

/opt/tw/bin/tripwire –interactive

Changed files/directories include:

EOF
cat
) | /bin/mailx -s "Tripwire file integrity report for `date`" $ADMINS
```

Appendix Q: /export/sysadmin/check_processes.sh

```
#!/sbin/sh
################################################################
#
# Name    : check_processes.sh
# Author  : John Worthing
# Date    : 5/24/2002
#
# This shell script will be run from cron every 5 minutes to see
# if named, sshd, or syslogd is running.  If one or more services
# are not detected then this script will start them and send an
# e-mail reporting it.
#
################################################################

ADMINS=jworthing@jii.com,admin2@jii.com,admin3@jii.com

/bin/ps -ef | /bin/grep named | grep -v grep # test for named

if [ $? = 0 ]; then

else

/opt/bind/sbin/named -u 1002 -t /chroot &
echo "named not running on `uname -n`.`domainname`" >
/tmp/named_running_test.mail
echo "Starting named on secondary named server." >>
/tmp/named_running_test.mail
echo "Date: `date`" >> /tmp/named_running_test.mail
echo "Command used: /opt/bind/sbin/named -u 1002 -t /chroot" >>
/tmp/named_running_test.mail
mailx -s "Output from named test" $ADMINS < /tmp/named_running_test.mail

fi

/bin/ps -ef | /bin/grep sshd | grep -v grep     # test for sshd

if [ $? = 0 ]; then

else

/usr/local/sbin/sshd
echo "sshd not running on `uname -n`.`domainname`" >
/tmp/sshd_running_test.mail
echo "Starting sshd on secondary sshd server." >> /tmp/sshd_running_test.mail
echo "Date: `date`" >> /tmp/sshd_running_test.mail
echo "Command used: /usr/local/sbin/sshd" >> /tmp/sshd_running_test.mail
mailx -s "Output from sshd test" $ADMINS < /tmp/sshd_running_test.mail

fi

/bin/ps -ef | /bin/grep syslogd | grep -v grep  # test for syslog

if [ $? = 0 ]; then

  exit
```

```
else

/usr/sbin/syslogd
echo "syslogd not running on `uname -n`.`domainname`" >
/tmp/syslogd_running_test.mail
echo "Starting syslogd on secondary syslogd server." >>
/tmp/syslogd_running_test.mail
echo "Date: `date`" >> /tmp/syslogd_running_test.mail
echo "Command used: /usr/sbin/syslogd" >> /tmp/syslogd_running_test.mail
mailx -s "Output from syslogd test" $ADMINS < /tmp/syslogd_running_test.mail

fi
```

Appendix R: /export/sysadmin/backup_slave.sh

```
#!/sbin/sh
################################################################
#
# Name    : backup_slave.sh
# Author  : John Worthing
# Date    : 9/21/2002
#
# This shell script will be run from cron weekly in order to
# backup key configuration files and logs that reside on ns2.
# Key files will be read from key_files.txt and then tarred
# and compressed to a date-named file.  The file is then copied
# via scp to a secure network server for archival.
#
################################################################

KEYFILE="/export/sysadmin/backups/key_files.txt";
TARFILE="/export/sysadmin/backups/`date '+%d%b%Y'`.tar";

/bin/tar cvf $TARFILE -I $KEYFILE;
/bin/compress $TARFILE;
/usr/local/bin/scp $TARFILE server:/archives/ns2_dir/;
```

Appendix S: Results of Test 1 -- Port Scans, Connections, and Processes

```
# nmap (V. 3.00) scan initiated Thu Oct 10 18:12:35 2002 as: nmap -sT -sU -sR
-P0 -O -oN ./nmap_scan_ns2.out 10.1.1.9

Interesting ports on ns2 (10.1.1.9):
(The 3063 ports scanned but not shown below are in state: closed)

Port       State       Service (RPC)
22/tcp     open        ssh
53/tcp     open        domain
53/udp     open        domain
32786/udp  open        sometimes-rpc26
32787/udp  open        sometimes-rpc28

No exact OS matches for host (If you know what OS is running on it, see
http://www.insecure.org/cgi-bin/nmap-submit.cgi).
TCP/IP fingerprint:
SInfo(V=3.00%P=sparc-sun-solaris2.7%D=10/10%Time=3DA61E25%O=22%C=1)
TSeq(Class=TR%IPID=RD%TS=100HZ)
T1(Resp=Y%DF=Y%W=60DA%ACK=S++%Flags=AS%Ops=NNTNWM)
T2(Resp=N)
T3(Resp=N)
T4(Resp=Y%DF=Y%W=0%ACK=O%Flags=R%Ops=)
T5(Resp=Y%DF=Y%W=0%ACK=S++%Flags=AR%Ops=)
T6(Resp=Y%DF=Y%W=0%ACK=O%Flags=R%Ops=)
T7(Resp=Y%DF=Y%W=0%ACK=S%Flags=AR%Ops=)
PU(Resp=Y%DF=Y%TOS=0%IPLEN=70%RIPTL=148%RIPCK=E%UCK=E%ULEN=134%DAT=E)

Uptime 0.042 days (since Thu Oct 10 17:40:49 2002)

Nmap run completed at Thu Oct 10 18:41:09 2002 -- 1 IP address (1 host up)
scanned in 1714 seconds

# netstat –a

UDP: IPv4
   Local Address        Remote Address     State
-------------------- -------------------- -------
      *.32786                            Idle
      *.32787                            Idle
      *.32788                            Idle
      *.32789                            Idle
localhost.domain                         Idle
ns2.domain                               Idle
      *.32798                            Idle
      *.32799                            Idle
      *.*                                Unbound

UDP: IPv6
   Local Address                 Remote Address                  State
If
--------------------------------- --------------------------------- ---------
- -----
      *.32799                                                    Idle

TCP: IPv4
```

```
   Local Address        Remote Address    Swind Send-Q Rwind Recv-Q State
-------------------- -------------------- ----- ------ ----- ------ -------
      *.*                  *.*                0      0 24576      0 IDLE
      *.22                 *.*                0      0 24576      0 LISTEN
      *.22                 *.*                0      0 24576      0 LISTEN
ns2.22            198.243.254.235.40132  8760     47 24820      0 ESTABLISHED
localhost.domain           *.*                0      0 24576      0 LISTEN
ns2.domain                 *.*                0      0 24576      0 LISTEN
localhost.953              *.*                0      0 24576      0 LISTEN
      *.*                  *.*                0      0 24576      0 IDLE

TCP: IPv6
   Local Address                        Remote Address                 Swind
Send-Q Rwind Recv-Q   State      If
-------------------------------- -------------------------------- ----- ---
--- ----- ------ ----------- -----
      *.*                                *.*                              0
0 24576      0 IDLE
      *.22                               *.*                              0
0 24576      0 LISTEN

Active UNIX domain sockets
Address  Type          Vnode     Conn  Local Addr      Remote Addr
70a19c28 stream-ord 00000000 00000000
70a19e68 stream-ord 00000000 00000000                (socketpair)

# ps -ef

     UID   PID  PPID  C   STIME TTY      TIME CMD
    root     0     0  0 15:44:28 ?       0:16 sched
    root     1     0  0 15:44:29 ?       0:00 /etc/init -
    root     2     0  0 15:44:29 ?       0:00 pageout
    root     3     0  0 15:44:29 ?      0:00 fsflush
    root   177     1  0 15:44:40 ?      0:00 /usr/lib/saf/sac -t 300
    root   180   177  0 15:44:41 ?      0:00 /usr/lib/saf/ttymon
    root    50     1  0 15:44:33 ?      0:00 /usr/lib/sysevent/syseventd
    root    52     1  0 15:44:33 ?      0:00 /usr/lib/sysevent/syseventconfd
    root   199   181  0 15:48:16 ?      0:00 /usr/local/sbin/sshd
    root   156     1  0 15:44:39 ?      0:00 /usr/sbin/cron
    root   151     1  0 15:44:39 ?      0:00 /usr/sbin/syslogd -t
    root   170     1  0 15:44:40 ?      0:00 /usr/lib/utmpd
    root   174     1  0 15:44:40 ?      0:00 /usr/lib/inet/xntpd
    root   178     1  0 15:44:40 console 0:00 -ksh
    root   167     1  0 15:44:40 ?      0:00 /usr/sbin/nscd
    root   181     1  0 15:44:41 ?      0:00 /usr/local/sbin/sshd
jworthin   203   201  0 15:48:19 pts/1   0:00 -ksh
jworthin   201   199  0 15:48:19 ?      0:00 /usr/local/sbin/sshd
    root   220   203  0 15:48:21 pts/1   0:00 ksh
    root   319   220  0 17:06:56 pts/1   0:00 ps -ef
   named   311     1  0 17:03:40 ?      0:00
   /opt/bind-9.2.2rc1/sbin/named -u named -t /chroot
```

Appendix T: Test 2 -- Alerts

Scenario for this test:

This inspiration for this demonstration came from a real life situation we had here recently with a developer on one of our boxes. This person required a wide range of permissions and flexibility in order to their job. This person was granted sudo access with full Admin privileges. It was later discovered that those privileges were being abused because this person was reading e-mails being sent by other users – members of that person's own development team actually. The following e-mail alert trail was generated as a test to simulate more or less what happened that day when this person was discovered. For the sake of this exercise, we'll refer to this person as Power_User.

Power_User attempts to su:

```
From:    Super-User [root@logserver]
Sent:    Friday, October 11, 2002 10:23 AM
To:      jworthing@jii.com,admin2@jii.com,admin3@jii.com
Subject: Message from Logserver Swatch

Oct 11 10:23:17 ns2 su: [ID 810491 auth.crit] 'su root' failed for root on
/dev/pts/9
```

Power_User attempts to use sudo to su:

```
From:    Power_User@logserver
Sent:    Friday, October 11, 2002 12:43 PM
To:      jworthing@jii.com,admin2@jii.com,admin3@jii.com
Subject: *** SECURITY information for ns2 ***

ns2 : Oct 11 12:43:19 : test : command not allowed ; TTY=pts/9 ;
PWD=/export/home/Power_User ; USER=root ; COMMAND=/usr/bin/su -
```

Power_User attempts to search for mail sent by me using grep:

```
From:    Super-User [root@logserver]
Sent:    Friday, October 11, 2002 10:47 AM
To:      jworthing@jii.com,admin2@jii.com,admin3@jii.com
Subject: Message from Logserver Swatch

Oct 11 10:46:03 ns2 /usr/local/bin/sudo: [ID 850335 local2.notice]
test : TTY=pts/9 ; PWD=/export/home/Power_User ; USER=root ;
COMMAND=/usr/bin/grep jworthin /var/spool/mqueue/*
```

Power_User realizes permissions are insufficient to use grep:

```
From:    Super-User [root@logserver]
Sent:    Friday, October 11, 2002 10:53 AM
To:      jworthing@jii.com,admin2@jii.com,admin3@jii.com
Subject: Message from Logserver Swatch

Oct 11 10:51:13 ns2 /usr/local/bin/sudo: [ID 850335 local2.notice]
test : TTY=pts/9 ; PWD=/export/home/Power_User ; USER=root ;
COMMAND=/usr/bin/ls -ld /var/spool/mqueue
```

Power_User uses sudo to change permissions on the mqueue directory:

```
From:    Super-User [root@logserver]
Sent:    Friday, October 11, 2002 10:56 AM
To:      jworthing@jii.com,admin2@jii.com,admin3@jii.com
Subject: Message from Logserver Swatch

Oct 11 10:55:11 ns2 /usr/local/bin/sudo: [ID 850335 local2.notice]
test : TTY=pts/9 ; PWD=/export/home/Power_User ; USER=root ;
COMMAND=/usr/bin/chmod 770 /var/spool/mqueue
```

Power_User searches again and this time is successful:

```
From:    Super-User [root@logserver]
Sent:    Friday, October 11, 2002 10:57 AM
To:      jworthing@jii.com,admin2@jii.com,admin3@jii.com
Subject: Message from Logserver Swatch

Oct 11 10:56:13 ns2 /usr/local/bin/sudo: [ID 850335 local2.notice]
test : TTY=pts/9 ; PWD=/export/home/Power_User ; USER=root ;
COMMAND=/usr/bin/grep jworthin /var/spool/mqueue/*
```

SSHD notification from wrappers generated during nmap scan:

```
From:    root@ns2.jii.com
Sent:    Friday, October 11, 2002 1:19 PM
To:      jworthing@jii.com,admin2@jii.com,admin3@jii.com
Subject: sshd- security-server.jii.com

[security-server.jii.com]
```

Appendix U: Test 3 results and some notes on security mechanisms in DNS

The following information on the history of security mechanisms in BIND as well as TSIG theory has been taken from an article written by BIND guru Cricket Liu called "Transactional Security in BIND 9" and can be read at http://www.linux-mag.com/2001-11/bind9_01.html. The important points have been regurgitated here in an effort to underscore the significance of the test results that will follow at the end.

Cricket Liu's article states the following about Transactional Security in BIND 9 with focus on TSIG theory and implementation:

> Back in BIND 4, the only security mechanism name servers supported was IP address-based access lists, and you could only use them to restrict zone transfers. The Internet Engineering Task Force (IETF) extended DNS to add security features, as the need for greater DNS security arose. In particular, the DNS Security Extensions (DNSSEC), described in RFC 2535 introduced cryptographic data integrity checking and source authentication to DNS. To provide these, DNSSEC uses asymmetric cryptography, better known as public key encryption.
>
> Unfortunately, while asymmetric cryptography is great for solving key distribution problems, it brings with it a big problem of its own; it's computationally intensive and consequently, fairly time-consuming. This means it is impractical for resolvers or dynamic updates. Applications that use resolvers need their queries processed as quickly as possible and servers that send or receive dynamic updates need to handle those updates promptly.
>
> The IETF recognized this shortcoming in DNSSEC and developed Transaction Signatures (TSIG), an alternate, lightweight security mechanism for use specifically by resolvers and dynamic updates and is codified in RFC 2845. Instead of slow asymmetric encryption, TSIG uses a relatively fast one-way hash function, making it quite suitable for use in even the most time-critical transactions and on the busiest name servers.
>
> TSIG takes advantage of some mathematical magic called a one-way hash function. One-way hash functions, also known as cryptographic checksums or message digests, calculate fixed-sized outputs from arbitrarily large inputs. In TSIG, a DNS query, response, or dynamic update is run through HMAC-MD5. A key, shared between the two endpoints of the transaction (e.g., between an updater and the name server receiving the update), is also used as input. The resulting hash value is placed in a new resource record called a TSIG record, which is added to the DNS message.
>
> The format of the TSIG record isn't really important; the record is a "meta-record," which is added to a DNS message automatically by the sender and stripped off and verified by the receiver. The verification of a TSIG record establishes two things: that a holder of the correct TSIG key signed the DNS message and that the

message wasn't modified after it was signed. To put it simply, a signer (or modifier) without the correct key will not be able to produce the right hash value.

So let's test TSIG to see if it is working as advertised on ns2. This can be accomplished quite easily using the BIND 9 version of dig. Here is an attempted zone transfer without a valid TSIG key:

```
# dig @10.1.1.9 axfr jii.com.

; <<>> DiG 9.2.2rc1 <<>> @10.1.1.9 axfr jii.com.
;; global options:  printcmd
; Transfer failed.
```

And here is a zone transfer from ns2 with a valid TSIG key:

```
# dig @10.1.1.9 axfr jii.com. -y rndc-key:fFln+QxfJkqai7l/3WDFAg==

; <<>> DiG 9.2.2rc1 <<>> @10.1.1.9 axfr jii.com. -y rndc-
key:fFln+QxfJkqai7l/3WDFAg==
;; global options:  printcmd
jii.com.    3600  IN SOA   ns1.jii.com. root.ns1.jii.com. 2002100701 10800
3600 2592000 86400
jii.com.    3600  IN NS ns2.jii.com.
jii.com.    3600  IN NS ns1.jii.com.
jii.com.    3600  IN A  10.1.1.1
jii.com.    3600  IN MX 5 juniper.jii.com.
www.jii.com.      3600  IN A  10.1.1.1
jii.com.    3600  IN SOA   ns1.jii.com. root.ns1.jii.com. 2002100701 10800
3600 2592000 86400
rndc-key.      0  ANY   TSIG  hmac-md5.sig-alg.reg.int. 1034096683 300 16
zBZ7Ud152cs1yKRzDVJkUw== 55842 NOERROR 0
;; Query time: 13 msec
;; SERVER: 10.1.1.9#53(10.1.1.9)
;; WHEN: Tue Oct  8 11:07:58 2002
;; XFR size: 8 records
```

So TSIG is working, which helps to secure zone transfers immensely. Now we need to provide a certain level of security for the DNS request itself. Unfortunately, the DNS message is transmitted in the clear so the best we can do is to limit access to these DNS messages to legitimate requests. Views in BIND 9 help to accomplish this by providing the concept of internal views vs. external views.

Most large domains contain host information for machines that are on both a private network and a public network. Many of these hosts, like web servers or DNS servers, have public identities since their purpose is to serve information to the Internet. Many more hosts yet, exist behind a firewall in a private network helping to support applications that run on these web servers. These in turn have strictly private identities and from a security perspective, should remain that way if possible.

Views allow you to do this in a simple way. An internal view is created that contains all the 10.1.1.x and 192.168.1.x hostnames in domain zone files as well as reverse lookup zones for the various internal subnets. An external view is created as well for these

domain zones that contains only the hostnames that are outward facing, and need to be resolvable from the outside world.

The configuration of ns2 allows for these views, so let's test to see if they are working. This is a DNS query from an internal host:

```
# nslookup www.jii.com
Server:    ns2.jii.com
Address:   10.1.1.9

Name:      www.jii.com
Address:   10.1.1.1
```

This is a DNS query from a legitimate external host:

```
# nslookup www.jii.com
Server:    ns2.jii.com
Address:   10.1.1.9

Name:      www.jii.com
Address:   198.62.160.125
```

This is a DNS query from a denied host:

```
# nslookup www.jii.com
*** Can't find server name for address 10.1.1.9: No response from server
*** Default servers are not available
```

Appendix V: Results of Test 4 -- Tripwire

```
added:   drwxr-xr-x root          512 Oct 11 09:27:47 2002 /dev/lp
deleted: -rw------- root           70 Oct  8 16:47:44 2002 /etc/ftpusers
changed: prw------- root            0 Oct 10 17:47:30 2002 /etc/cron.d/FIFO
changed: -rw-r--r-- root          314 Oct 10 17:47:22 2002 /etc/coreadm.conf
changed: -rw-r--r-- root          236 Oct 10 17:47:29 2002 /etc/dumpadm.conf
changed: -r--r--r-- root          742 Oct 11 09:26:31 2002 /etc/passwd
changed: -r-------- root           31 Oct 11 09:27:24 2002 /etc/shadow
changed: -rwxr-xr-x root        20825 Oct 10 17:51:31 2002 /usr/sbin/tar
### Attr        Observed (what it is)        Expected (what it should be)
### ========== ============================= =============================
/etc/cron.d/FIFO
      st_mtime: Thu Oct 10 17:47:30 2002      Thu Oct 10 17:32:39 2002
      st_ctime: Thu Oct 10 17:47:30 2002      Thu Oct 10 17:32:39 2002

/etc/coreadm.conf
      st_mtime: Thu Oct 10 17:47:22 2002      Thu Oct 10 17:40:59 2002
      st_ctime: Thu Oct 10 17:47:22 2002      Thu Oct 10 17:40:59 2002

/etc/dumpadm.conf
      st_mtime: Thu Oct 10 17:47:29 2002      Thu Oct 10 17:32:39 2002
      st_ctime: Thu Oct 10 17:47:29 2002      Thu Oct 10 17:32:39 2002

/etc/passwd
       st_size: 742                           715
      st_mtime: Fri Oct 11 09:26:31 2002      Thu Oct 10 14:24:52 2002
      st_ctime: Fri Oct 11 09:26:31 2002      Thu Oct 10 14:24:52 2002
    md5 (sig1): 0OZ67xPNv2gE1KdyPzr9sh        2bejXvo54B0EQ01YPPx9SH
 snefru (sig2): 3VOsHQ8YDMyYczqTXT0xCE        2CCeR2M8FbNe8uyVZF39F2

/etc/shadow
       st_size: 31                            335
      st_mtime: Fri Oct 11 09:27:24 2002      Wed Oct  9 17:01:13 2002
      st_ctime: Fri Oct 11 09:27:24 2002      Wed Oct  9 17:01:13 2002
    md5 (sig1): 1rMt7DAgY5.Q9MH57zS:Rf        0eYQuvqocr0JAai3y:KpWo
 snefru (sig2): 1OFoNsG8u7x5Ws8UqA8bEM        095PaxTaU17:R2C.VfZNHL

/usr/sbin/tar
       st_mode: 100755                        100555
        st_ino: 449164                        448992
        st_gid: 1                             2
       st_size: 20825                         66252
      st_mtime: Thu Oct 10 17:51:31 2002      Thu Aug 29 16:15:09 2002
      st_ctime: Thu Oct 10 17:53:08 2002      Wed Oct  9 08:37:33 2002
    md5 (sig1): 0Uop0gpsUi9BfAQB:rfLLA        0WKuHASMTSFj7Y4CTyJPUW
 snefru (sig2): 11IxoKdGjocEIbR44c8GDW        1QKE9EFOr8d3iedbHYJ8GD
```

Appendix W: Results of Test 5 – Nessus Security Scanner

```
Nessus Scan Report
------------------

SUMMARY

 - Number of hosts which were alive during the test : 1
 - Number of security holes found : 0
 - Number of security warnings found : 2
 - Number of security notes found : 6

TESTED HOSTS

 ns2 (Security warnings found)

DETAILS

+ ns2 :
 . List of open ports :
   o unknown (22/tcp) (Security warnings found)
   o domain (53/tcp)
   o unknown (1241/tcp) (Security warnings found)

 . Warning found on port unknown (22/tcp)

    The remote SSH daemon supports connections made
    using the version 1.33 and/or 1.5 of the SSH protocol.

    These protocols are not completely cryptographically
    safe so they should not be used.

    Solution :
     If you use OpenSSH, set the option 'Protocol' to '2'
     If you use SSH.com's set the option 'Ssh1Compatibility' to 'no'

    Risk factor : Low

 . Information found on port unknown (22/tcp)


    An ssh server is running on this
     port

 . Information found on port unknown (22/tcp)


    Remote SSH version : SSH-1.99-OpenSSH_3.4p1

 . Information found on port unknown (22/tcp)


    The remote SSH daemon supports the following versions of the
    SSH protocol :

      . 1.33
      . 1.5
```

```
      . 1.99
      . 2.0

   . Warning found on port unknown (1241/tcp)


      A Nessus Daemon listens on this port.
      supported versions: < NTP/1.0 >< NTP/1.1 >< NTP/1.2
       >


   . Information found on port unknown (1241/tcp)


      A TLSv1 server answered on this port

   . Information found on port unknown (1241/tcp)


      Here is the TLSv1 server certificate:
      Certificate:
          Data:
              Version: 3 (0x2)
              Serial Number: 1 (0x1)
              Signature Algorithm: md5WithRSAEncryption
              Issuer: C=US, ST=CO, L=Denver, O=JII, OU=Certification Authority
for
       ns2, CN=ns2/Email=ca@ns2
              Validity
                  Not Before: Oct  9 23:10:48 2002 GMT
                  Not After : Oct  9 23:10:48 2003 GMT
              Subject: C=US, ST=CO, L=Denver, O=JII, OU=Server certificate for
       ns2, CN=ns2/Email=nessusd@ns2
              Subject Public Key Info:
                  Public Key Algorithm: rsaEncryption
                  RSA Public Key: (1024 bit)
                      Modulus (1024 bit):
                          00:a7:ed:9e:15:9e:98:cb:06:19:7e:6e:14:a3:24:
                          37:2b:6c:8b:6f:05:5a:ab:2e:48:f8:59:b3:20:ea:
                          f9:26:38:e5:c7:64:90:09:b5:92:a3:fe:e9:74:57:
                          78:e2:b3:7b:72:c7:12:a5:db:35:63:97:65:ba:86:
                          51:b7:d4:57:a0:87:08:72:47:bc:39:bf:4a:44:5a:
                          36:c4:36:00:e1:8a:21:6d:0a:32:27:f5:97:95:82:
                          07:35:af:d4:73:63:9f:30:09:7b:df:cf:78:63:4b:
                          8d:6c:44:7f:f0:a4:04:eb:b7:da:87:18:05:bf:22:
                          a1:96:67:c8:0d:0f:dc:96:a7
                      Exponent: 65537 (0x10001)
              X509v3 extensions:
                  Netscape Cert Type:
                      SSL Server
                  X509v3 Key Usage:
                      Digital Signature, Non Repudiation, Key Encipherment
                  Netscape Comment:
                      OpenSSL Generated Certificate
                  X509v3 Subject Key Identifier:

9D:E5:D9:CA:F2:AE:D5:BD:0B:A7:D6:AF:53:B2:2C:50:12:10:66:36
```

```
                X509v3 Authority Key Identifier:

      keyid:73:0F:32:E7:2F:A8:15:1A:7B:C1:26:1D:2F:A1:85:23:37:9D:09:F6
                    DirName:/C=US/ST=CO/L=Denver/O=JII/OU=Certification
      Authority for ns2/CN=ns2/Email=ca@ns2
                    serial:00

                X5 09v3 Subject Alternative Name:
                    email:nessusd@ns2
                 X509v3 Issuer Alternative Name:
                    <EMPTY>

           Signature Algorithm: md5WithRSAEncryption
               34:1e:65:00:a6:cf:2f:b0:0d:5f:aa:dc:ec:64:ca:21:32:b5:
               89:79:da:9b:31:1a:09:f2:06:1f:2a:93:41:fe:53:53:00:fa:
               c5:46:f2:84:d1:a5:49:a2:0d:32:ac:f9:16:96:3d:a4:fa:83:
               35:e0:a2:d6:ac:86:a9:44:87:d1:2a:e2:3d:5c:60:d5:14:4a:
               dd:62:2d:5a:a3:1a:43:ce:53:0d:b5:bb:ee:7e:37:fe:d4:df:
               af:1b:e2:6a:12:d8:b5:6b:60:cb:84:ab:c9:f2:56:d1:41:15:
               9d:6d:1e:f2:3b:3b:0a:e6:d9:22:e2:09:5c:f2:5f:c4:43:1e:
               f1:71

 . Information found on port unknown (1241/tcp)


    This TLSv1 server does not accept SSLv2 connections.
    This TLSv1 server does not accept SSLv3 connections.


----------------------------------------------------
This file was generated by the Nessus Security Scanner
```

List of References

Internet Sources:

Liu, Cricket. "Transactional Security in BIND 9." November 2001. URL: http://www.linux-mag.com/2001-11/bind9_01.html.

Martin, Derek D. "Securing BIND: How To Prevent Your DNS Server from Being Hacked." 21 May 2001. URL: http://rr.sans.org/DNS/sec_BIND.php.

SANS Institute Publications. "Solaris Security: Step-by-Step Table of Contents." 1999. URL: http://www.sans.org/newlook/publications/solaris_toc.htm.

Boran, Seán. "Comparison of Solaris Hardening Scripts." 24 November 2000. URL: http://www.netsecurity.pl/www.boran.com/security/sp/comparison1.html.

Sabernet.net. "Solaris Security Guide." URL: http://sabernet.home.attbi.com/papers/Solaris.html.

Magazine Articles:

Widdowson, Liam. "Jailed Internet Services." Sys Admin. Volume 10, Number 8 (August 2001): pp 39-45.

Laudicina, Alan P. "Nessus – A Powerful, Free Remote Security Scanner." Sys Admin. Volume 11, Number 5 (May 2002): pp 24-30.