



# Global Information Assurance Certification Paper

Copyright SANS Institute  
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permitted without express written permission.

---

# Solaris 8 (sparc) Security Checklist for JFY, Inc.

This document defines the Security Checklist for the TotalApps product of JFY, Inc. It assumes a familiarity with the Sun Solaris Sparc products including all aspects of System Administration and Installation. It assumes that the TotalApps product is to be installed after the initial Sun instation. The basic philosophy followed is one of "defense in depth." Even features that aren't currently used are configured in their most secure state-- for example, if the application requires a particular port be enabled, that can be done in a secure way at a later time.

The following conventions are used to make this document easier to follow.

1. Commands to be executed or shell scripts are in a `red courier` typeface:

```
echo 172.16.0.1 > /etc/defaultrouter
```

2. Results from a command are indicated in a `courier green` typeface:

```
ls -lL /etc/rc2.d/S74new${FILE}
### <<< Check the results: >>>
-rwx----- 1 root root 914 May 4 15:22 /etc/rc2.d/S74newsyslog
```

3. File contents are indicated in a `courier blue` typeface:

```
# who I send unqualified names to ...
DRmailhost

# who gets all local email traffic ...
DHmailhost
```

Quick start information:

\_\_\_\_ Unpack the System insert the most current Solaris CDROM and apply power (DO NOT connect the system to a network until it is fully secured).

Note the Solaris Version installed here: \_\_\_\_\_ (e.g. Solaris 8 01/01).

\_\_\_\_ Make sure that the system is completely physically secure during this entire installation process.

\_\_\_\_ **DO NOT** connect the system to any network [\[1\]](#); connecting the main Ethernet port to an unconnected hub will speed up installation. Do not connect the system to any network until all the following steps have been completed. ([More...](#))

\_\_\_\_ [\*\*Pre-installation Requirements\*\*](#)

\_\_\_\_ [\*\*Sun Installation\*\*](#)

\_\_\_\_ [\*\*Application Installation\*\*](#)

\_\_\_\_ [\*\*Post-installation Configuration\*\*](#)

_____	<a href="#"><u>Additional recommended packages</u></a>
_____	<a href="#"><u>Sun patches</u></a>
_____	<a href="#"><u>Configuring Third Party Applications</u></a>
_____	<a href="#"><u>Network Service Restrictions</u></a>
_____	<a href="#"><u>File System Configuration</u></a>
_____	<a href="#"><u>Account Administration</u></a>
_____	<a href="#"><u>Solaris Auditing Features</u></a>
_____	<a href="#"><u>Miscellaneous Security Enhancements</u></a>
_____	<a href="#"><u>Openboot Security</u></a>
_____	<a href="#"><u>Testing</u></a>
_____	<a href="#"><u>System Backups</u></a>
_____	<a href="#"><u>Physical Security</u></a>
_____	<a href="#"><u>REFERENCES</u></a>
_____	<a href="#"><u>APPENDIX</u></a>

## [Pre-installation Requirements](#)

### Preinstallation Information

#### Basic System Description

Define the SPARC system type, memory size and critical parameters.

System type:	_____	(Ultra 250, 450, 3500, etc.)
Memory size (MB)	_____	
Processor Speed:	_____	
Number of Processors:	_____	
System Serial #:	_____	

HostID: \_\_\_\_\_

System name: \_\_\_\_\_

Domain name: \_\_\_\_\_

IP Address(es) : \_\_\_\_\_  
(and net masks) \_\_\_\_\_

The use of DHCP is not recommended for an Internet connected system.

## File Systems and partitions

Define the disks available and the partitions required by the application.

### Drive c0t0d0

<u>Slice</u>	<u>Name</u>	<u>Suggested Use</u>
0: _____		Root File System (120MB)
1: _____		/var File System (200MB)
2: _____		Swap File System (Size: same as physical memory)
3: _____		spare
4: _____		spare
5: _____		spare
6: _____		/usr File System (170 MB)
7: _____		/local (remaining space)

This allows more flexibility in mounting whole file systems "read only." If Sun's DiskSuite (mirroring) is to be used, then one partition per disk is needed for the metadb information.

[Previous](#)   [Index](#)   [Top](#)

## [Sun Installation](#)

Follow the specific instructions for the Solaris installation of the product. The following guidelines are recommended [\[2\]](#):

- \_\_\_\_\_ Verify that the system is not connected to any network ([More..](#)).
- \_\_\_\_\_ Boot from the most recent Solaris OS CD-ROM -- (the one marked "Solaris 8 SOFTWARE 1 of 2):  

```
boot cdrom - - w      ### NOTE the space between '-' and 'w'!
```

 (this speeds things up by running without webstart; disconnecting the keyboard and loading via ttya is even faster).
- \_\_\_\_\_ Supply the language and possibly terminal information as appropriate for your installation.
- \_\_\_\_\_ Enter the "host name" from the [Pre-installation](#) sheet.

- \_\_\_\_\_ Select "Networked."
- \_\_\_\_\_ Select "No" for DHCP (DHCP is *not* recommended, [more...](#)).
- \_\_\_\_\_ Define the IP Address.
- \_\_\_\_\_ Select "No" for IPV6 [\[3\]](#).
- \_\_\_\_\_ Select "No" for Kerberos (this must be enabled later since there is no network connection).
- \_\_\_\_\_ Select "None" for name service (this can easily be defined later).
- \_\_\_\_\_ Define the netmask if it is not a standard mask.
- \_\_\_\_\_ Select the correct time zone.
- \_\_\_\_\_ Verify the correct system date and time.

The preferred installation from a security perspective is to install the bare minimum. There may be specific application requirements that require more components and/or services; these can be added as required after first securing the system [\[4\]](#). ([More...](#))

- \_\_\_\_\_ Choose "Initial" install to begin with a clean slate for the system. Not only is this much faster than an upgrade, it assures that nothing from a previous installation will be left over that could compromise the security of the system.
- \_\_\_\_\_ Choose "Standalone" server.
- \_\_\_\_\_ Select "Core System Support." If not selected, note exception here: \_\_\_\_\_
- \_\_\_\_\_ Define the file system layout on the disks. The standard SUN file systems for ROOT (/), /usr and /var should be defined separately. In addition, you may need to define /opt or /local for additional packages that you install. If SUN's DiskSuite (mirroring) is used, then you need one spare partition per drive for the metadb.
- \_\_\_\_\_ Don't mount any remote file systems.
- \_\_\_\_\_ Choose the auto-reboot option.
- \_\_\_\_\_ When the system reboots after installation (about five to ten minutes) define a [secure ROOT password](#).

[Previous](#)   [Index](#)   [Top](#)

## [Post-installation Configuration](#)

- \_\_\_\_\_ Disable files in /etc/rc?.d that are not required by moving them to the name ".NOSnn.xxx" [\[5\]](#):

```
cd /etc/rc2.d
for file in \
    sysid.net ldap.client rpc sysid.sys autoinstall \
    cachefs.daemon nfs.client autofs nscd PRESERVE \
    sendmail cacheos.finish nfs.server
do
    name=`echo S??${file}`
    mv $name .NOS${name}
```

done

\_\_\_\_ Configure sendmail for no local mail delivery by changing the "DR" directive in the /etc/mail/sendmail.cf file following the example file in [\[6\]](#):

```
# who I send unqualified names to ...
DRmailhost

# who gets all local email traffic ...
DHmailhost

#!/sbin/sh
ed - <<-!EOF!
/^DR$/ s//DRmailhost/
/^DH$/ s//DHmailhost/
w
w
q
!EOF!
```

\_\_\_\_ Change the ROOT crontab(use EDITOR=vi crontab -e) to run sendmail -q, only to cleanup the queue:

```
12 * * * * /usr/lib/sendmail -q
```

\_\_\_\_ Disable the SUN dynamic routing protocol by defining /etc/defaultrouter according to the local requirements. If dynamic routing is required, use the gated daemon found at <http://www.gated.org>.

```
echo 172.16.3.1 > /etc/defaultrouter
```

\_\_\_\_ Although it is redundant, it is a good idea to define the /etc/notrouter file (unless dynamic routing is required):

```
touch /etc/notrouter
```

\_\_\_\_ Add entries to /etc/resolv.conf file for the local DNS hosts:

```
domain jfy.com _____
nameserver 172.16.1.10 _____
nameserver 172.16.1.9 _____
```

\_\_\_\_ Append the keyword "dns" the /etc/nsswitch.conf file to include DNS for finding hosts:

```
hosts: files dns
```

\_\_\_\_ Setup the /etc/defaultdomain file:

```
echo jfy.com > /etc/defaultdomain _____
domainname jfy.com
```

\_\_\_\_ Add the fully qualified system name to the /etc/hosts file:

```
172.16.3.111    myname    myname.jfy.com    loghost
```

\_\_\_\_\_ Control the umask setting in the file /etc/default/init at all run levels of the system to 022 or better so that no files with group or world write privileges are created:

```
CMASK=022
```

\_\_\_\_\_ If this host is not being used as a logging host for other hosts, then run syslogd with the -t flag [\[7\]](#):

```
#!/sbin/sh
FILE=syslog
cd /etc/init.d
rm -f /etc/rc2.d/S74${FILE} /etc/rc2.d/S74new${FILE}
cp ${FILE} new${FILE}
ed - <<!EOF! new${FILE}
/^.*\usr\sbin\syslogd / s//&-t /
w
w
q
!EOF!
chmod 700 /etc/init.d/new${FILE}
chown root:root /etc/init.d/new${FILE}
ln -s /etc/init.d/new${FILE} /etc/rc2.d/S74new${FILE}
ls -lL /etc/rc2.d/S74new${FILE}
### <<< Check the results: >>>
-rwx----- 1 root root 914 May 4 15:22 /etc/rc2.d/S74newsyslog
```

\_\_\_\_\_ If auto "pty" allocations or "hot pluggable" hardware devices are required, then skip this step. Follow the recommendation in Solaris Security [\[8\]](#) to disable the devfsadm and devfseventd daemons:

```
#!/sbin/sh
cd /etc/init.d
rm -f /etc/rc[2S].d/S50*devfsadm
cp devfsadm newdevfsadm
ed - <<!EOF! newdevfsadm
1,$ g/devfsadm/ s/^/### /
1,$ g/devfseventd/ s/^/### /
w
w
q
!EOF!
chmod 700 /etc/init.d/newdevfsadm
chown root:root /etc/init.d/newdevfsadm
ln -s /etc/init.d/newdevfsadm /etc/rc2.d/S50newdevfsadm
ln -s /etc/init.d/newdevfsadm /etc/rc2.d/S50newdevfsadm
ls -lL /etc/rc[2S].d/S50*devfsadm
### <<< Check the results: >>>

-rwx----- 1 root root 1392 May 4 15:24 /etc/rc2.d/S50newdevfsadm
-rwx----- 1 root root 1392 May 4 15:24 /etc/rcS.d/S50newdevfsadm
#
```

\_\_\_\_\_ Unless IPV6 is a requirement, disable the in.lpd daemon:

```
mv /usr/lib/inet/in.ndpd /usr/lib/inet/in.ndpd.rls
```

[Previous](#) [Index](#) [Top](#)

## Application Installation

There are a number of useful applications that are not installed as a part of the "Core System Support" and these should be installed [\[9\]](#):

\_\_\_\_\_ Be sure that the SUN Volume manager is not installed by removing the packages:

```
pkgrm SUNWvolu
pkgrm SUNvolr
pkgrm SUNWvolg
```

\_\_\_\_\_ Place the SUN CDROM (the one that says "SOLARIS 8 SOFTWARE, 1 of 2") media in the drive.

\_\_\_\_\_ Mount CDROM on /mnt (This assumes that the normal Sun "vold" is not running; the actual device can probably be found by checking names in /dev/dsk):

```
mkdir /tmp/mnt
mount -r -F hsfs /dev/dsk/c0t6d0s0 /tmp/mnt
```

\_\_\_\_\_ Change to the product directory:

```
cd /tmp/mnt/Solaris_8/Product
```

In the following sections, the pkgadd command will require numerous "yes" responses in order to complete the installation. This can be simplified by creating a response file in /var/tmp/a and using the following command:

```
pkgadd -a /var/tmp/a </dev/null -d . <<<arguments from below>>>
```

where the file /var/tmp/a is setup as follows:

```
cat << !EOF! > /var/tmp/a
mail=
instance=unique
partial=nocheck
runlevel=nocheck
idepend=nocheck
rdepend=nocheck
space=nocheck
setuid=nocheck
conflict=nocheck
action=nocheck
basedir=default
!EOF!
```

\_\_\_\_\_ In order to install and check the installed patches, the following packages must be added (these are required by the SUN supplied patchdiag or patchchk scripts and can probably be removed after patches are checked if so desired):



```
pkgadd -a /var/tmp/a </dev/null -d . \
    SUNWpl5u SUNWadmfw SUNWadmap SUNWadmC SUNWlibC \
    SUNWesu SUNWswmt
```

\_\_\_\_ Verify that the `showrev -p` command now works.

\_\_\_\_ Add the Berkely tools and SUN documentation tool packages [\[10\]](#):

```
pkgadd -a /var/tmp/a </dev/null -d . SUNWdoc SUNWscpr SUNWscpu
```

\_\_\_\_ Add the [NTP](#) Time server, Berkely tools and [documentation](#) packages [\[11\]](#):

```
pkgadd -a /var/tmp/a </dev/null -d . SUNWntpr SUNWntpu
```

\_\_\_\_ Unmount the Solaris install CDROM and remount the second SUN INSTALLATION CDROM 2 of 2:

```
cd /
umount /tmp/mnt
eject /dev/dsk/c0t6d0s0
```

\_\_\_\_ Use the `pkgadd` command to install the appropriate packages including [documentation](#) and `zlib` files (needed for `OPENssh`, remount the CDROM with the SUN INSTALLATION CDROM 2 of 2):

```
mount -r -F hsfs /dev/dsk/c0t6d0s0 /tmp/mnt
cd /tmp/mnt/Solaris_8/Product
pkgadd -a /var/tmp/a </dev/null -d . SUNWter SUNWaccr SUNWaccu \
    SUNWsprot SUNWbtool SUNWman SUNWzlib
```

\_\_\_\_ Note any other specific packages that are installed here ([be very careful what you install...](#)):

```
pkgadd -d . _____
```

\_\_\_\_ Unmount the CDROM:

```
cd /
umount /tmp/mnt
eject /dev/dsk/c0t6d0s0
```

[Previous](#) [Index](#) [Top](#)

## [Additional recommended packages](#)

There are a number of packages available that can substantially increase the security of the system. Remember that the security of the system is not a static condition, but an ongoing process. These applications will assist in maintaining a secure system as time goes on. Since there is no compiler installed on the system (why make the intruder's job any easier? [\[12\]](#)) these packages need to be built on another system. As an alternative, the compiler can be temporarily installed to build the applications and then removed. All these applications will all easily fit on a removable CDROM media. It's much safer to pre-configure these applications on a development system, copy them to removable media and then run only the final "make install" on the target system (rather than connecting the network):

\_\_\_\_ Obtain the TCP Wrappers source code from (use the `ipv6` version for Solaris 8):

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

```
tcp_wrappers_7.6-ipv6.1.tar.gz
tcp_wrappers_7.6-ipv6.1.tar.gz.sig
```

\_\_\_\_\_ If RPC must be used, obtain the source code from (use the ipv6 version for Solaris 8):

```
rpcbind_2.1tar.gz
rpcbind_2.1tar.gz.sig
```

\_\_\_\_\_ Also obtain the corresponding .sig file and check the signature with PGP (or GPG) tools (the actual version may be different):

```
gpg --verify tcp_wrappers_7.6-ipv6.1.tar.gz.sig
```

\_\_\_\_\_ Unpack the sources:

```
gunzip -c tcp_wrappers_7.6.tar.gz | tar -xf -
```

\_\_\_\_\_ Modify the top-level Makefile:

```
chmod 644 Makefile
vi Makefile
```

\_\_\_\_\_ Change the variables REAL\_DAEMON\_DIR and FACILITY (use the "advanced" installation option):

```
REAL_DAEMON_DIR=/usr/sbin
FACILITY= LOG_AUTH
```

\_\_\_\_\_ Build the software (make sure to set the CC=gcc if the GNU C compiler is being used):

```
make sunos5
```

\_\_\_\_\_ Install the files with the script given in [\[13\]](#):

```
mkdir -p /usr/local/include /usr/local/lib \
    /usr/local/bin /usr/local/sbin
for file in safe_finger tcpd tcpdchk tcpdmatch try-from
do
    /usr/sbin/install -s -f /usr/local/sbin \
        -m 0555 -u root -g daemon $file
done
/usr/sbin/install -s -f /usr/local/lib \
    -m 0444 -u root -g root libwrap.a
/usr/sbin/install -s -f /usr/local/include \
    -m 0444 -u root -g root tcpd.h
mkdir /etc/banners
cp Banners.Makefile /etc/banners/Makefile
cat <<-!EOF! > /etc/banners/prototype
Access to this system is controlled and restricted to authorized
JFY Users only. If you are not authorized specifically by JFY
then log off immediately. All transactions are subject to
monitoring.
!EOF!
```

```

cd /etc/banners
#### <<< Edit the Makefile for any local needs >>>
#### e.g. since there is no C compiler, change the definition
#### for 'nul:'
### nul:
###     echo '#!/sbin/sh\nexit 0\n' > nul
###     chmod 755 nul

make

```

The ssh (or OPENssh) package is extremely useful in keeping the system secure. This package can replace the functions of telnet and ftp, which send clear-text passwords over the net. The OPENssh package requires two library packages to be installed first, "zlib" which was installed from the Solaris CDROM above, and "OPENSSL." This section is adapted from [14].

\_\_\_\_\_ Obtain the OPENSSL package from:

<ftp://ftp.openssl.org/source/openssl-0.9.6a.tar.gz>

\_\_\_\_\_ Unzip and untar the distribution files (the specific version may be different):

```

gunzip -c openssl-0.9.6a.tar.gz | tar -xf -
cd openssl-*

```

\_\_\_\_\_ Configure, make and install the library (this will install the files in /usr/local):

```

sh ./config; make

```

\_\_\_\_\_ Copy these files to the target system (/usr/local/src/openssl-0.9.6a) and install using the command:

```

cd /usr/local/src/openssl-0.9.6a
/usr/ccs/bin/make install

```

\_\_\_\_\_ Obtain the OPENSSH package from:

[www://www.openssh.com/portable.html](http://www.openssh.com/portable.html)

\_\_\_\_\_ Unzip and untar the *portable* distribution (the version numbers may be different):

```

gunzip -c openssh-2.5.2p2.tar.gz | tar -xf -
cd openssh-2.5.2p2

```

\_\_\_\_\_ Configure, and make the package (this will prepare to install the files in /usr/local):

```

sh ./configure --prefix=/usr/local --with-tcp-wrappers \
    --without-rsh --disable-suid-ssh
make

```

\_\_\_\_\_ Use the package build tools in the Solaris install area:

```

cd contrib/solaris; ./build-pkg

```

\_\_\_\_\_ Copy the file to the target system and install it with pkgadd:

```

cd build-SSH-package

```

```
pkgadd -a /var/tmp/a < /dev/null -d . OPENssh
/etc/init.d/sshd stop
ln -s /usr/local/etc/ssh_host_key \
    /usr/local/etc/ssh_host_rsa_key
/etc/init.d/sshd start
```

\_\_\_\_ Obtain the Tripwire package from (the original Academic Source Release (ASR) distribution can be obtained by registering with Tripwire):

<http://www.tripwire.com>

\_\_\_\_ Unzip and untar the distribution:

```
gunzip -c tripwire-*.tar.gz | tar -xf -
cd tripwire*
```

\_\_\_\_ Configure, make and install the library (this will install the files in /usr/local):

```
sh ./configure
make
mkdir /usr/man/man5 /usr/man/man8
make INSTALL=/usr/ucb/install install
```

\_\_\_\_ If mailer functionality is required, then obtain the qmail package from <http://www.qmail.org>. Configure it according to the detailed step-by-step instructions provided for the service which is required.

\_\_\_\_ Download and the fix-modes package [\[24\]](#):

<ftp://ftp.fwi.uva.nl/pub/solaris/fix-modes.tar.gz>

\_\_\_\_ Build the fix-modes package:

```
make CC=gcc
```

[Previous](#) [Index](#) [Top](#)

## [Sun patches](#)

\_\_\_\_ Create the patch directory:

```
mkdir /var/tmp/patches
```

\_\_\_\_ Change to the patch directory:

```
cd /var/tmp/patches
```

\_\_\_\_ Obtain the SUN recommended patches, including all recommended Security patches. The patch files may be obtained from [ftp://sunsolve1.sun.com/patches/8\\_recommended.zip](ftp://sunsolve1.sun.com/patches/8_recommended.zip):

```
ftp sunsolve1.sun.com
<login as user "ftp", use your email address as password>
ftp> bin
ftp> cd /patches
ftp> get 8_Recommended.zip
```

\_\_\_\_ Obtain the SUN patchdiag cross reference and checksums file. The files may be obtained from <ftp://sunsolve1.sun.com/patches/patchdiag.xref>.

```
ftp> get patchdiag.xref
```

\_\_\_\_ Obtain the SUN checksums file. The file may be obtained from <ftp://sunsolve1.sun.com/patches/CHECKSUMS>.

```
ftp> get CHECKSUMS
ftp> quit
```

\_\_\_\_ Check the patch checksums using the md5 program against the value listed in the CHECKSUMS file.

```
# sed < CHECKSUMS -n -e '/8_Recommended.zip/,/^$/ p'
8_Recommended.zip
MD5: 24329cbbade73eb2e43ceb50886bb0a3
SysV Sum: 22463 76206
Sum: 30604 76206

# md5 8_Recommended.zip;sum 8_Recommended.zip;sum -r
8_Recommended.zip
MD5 (8_Recommended.zip) = 24329cbbade73eb2e43ceb50886bb0a3
22463 76206 8_Recommended.zip
30604 76206 8_Recommended.zip
```

\_\_\_\_ Unzip the SUN recommended patches (the -qq flag may be desirable to quiet the noisy output of this command):

```
unzip -qq 8_Recommended.zip
```

\_\_\_\_ Install the SUN recommended patches (the -nosave option saves disk space but means that patches cannot be backed out, -q means "quiet" [more...](#)):

```
cd 8_Recommended
./install_cluster -nosave -q
```

\_\_\_\_ Obtain the patch check tool from <http://sunsolve.sun.com> and unzip it (this is similar to patchdiag available to SUN contract customers, either tool may be used):

```
zcat pchk_1.1.tar.Z | tar -xf -
```

\_\_\_\_ Run the patchk.pl file:

```
perl ./patchcheck_1.1/patchk.pl > /tmp/patches.txt
```

\_\_\_\_ Review the /tmp/patches.txt output file; it is not necessary for ALL patches to be at their current level, but any patches listed in the "Security Patches" section should be obtained from Sunsolve and installed. If any SUN packages are installed after this point, the steps in this section should be repeated.

[Previous](#) [Index](#) [Top](#)

## [Configuring Third Party Applications](#)

\_\_\_\_ Create /etc/hosts.allow and /etc/hosts.deny (more advanced features can be enabled with the -DPROCESS\_OPTIONS flag):

```
cat > /etc/hosts.allow <<-!EOF!
    sshd : 172.25.30.0/255.255.254.0 172.25.134.0/255.255.255.0
172.25.136.0/255.255.254.0 : banners /etc/banners : ALLOW
!EOF!

cat > /etc/hosts.deny <<-!EOF!
    ALL : ALL : spawn (/usr/sbin/safe_finger -l %@h | /bin/mailx -s
"Port Denial not
ed %d-%h" root) &
!EOF!
```

\_\_\_\_ Configuring tripwire's configuration files and building the initial database should be done according to the package instruction as the last step (see below).

## [Previous](#) [Index](#) [Top](#)

## [Network Service Restrictions](#)

\_\_\_\_ The simplest way to turn off all network services is to disable inetd. Even if this is done, unwanted services should still be disabled. To turn off inetd, install the simplified /etc/init.d/newinetsvc file from [\[15\]](#), reprinted in [Appendix INET](#):

```
#!/sbin/sh
rm -f /etc/rc?.d/S72inetsvc /etc/rc2.d/S72newinetsvc
cp ./newinetsvc /etc/init.d
chmod 700 /etc/init.d/newinetsvc
chown root:root /etc/init.d/newinetsvc
ln -s /etc/init.d/newinetsvc /etc/rc2.d/S72newinetsvc
```

\_\_\_\_ Disable all non-essential network services by inserting comment (#) characters before each unwanted service in /etc/inet/inetd.conf, note which services are active below:

---

---

---

For example, to preserve only the Telnet and FTP services, use the following commands (between the two [] brackets, type the <space> and <tab> characters):

```
cd /etc/inet
mv inetd.conf inetd.conf.full
touch inetd.conf      ### RECOMMENDED-- DISABLE ALL SERVICES ###

#### <<< -- OR -- NOT RECOMMENDED!!!: >>>
egrep '^(ftp|telnet)[      ]' > /etc/inet/inetd.conf
```

Be sure to change to the /etc/inet directory first, since /etc/inetd.conf is a symbolic link and unexpected results may occur if this is not done exactly as shown above.

[Previous](#) [Index](#) [Top](#)

## [File System Configuration](#)

\_\_\_\_\_ Mount /usr readonly by editing /etc/vfstab:

```
/dev/dsk/c0t0d0s4 /dev/rdisk/c0t0d0s4 /usr ufs 1 no ...,ro
```

\_\_\_\_\_ Mount /var, /tmp and /local with nosuid attributes by editing /etc/vfstab (ROOT can't be mounted nosuid because this also means nodev):

```
/dev/dsk/c0t0d0s5 /dev/rdisk/c0t0d0s5 /var ufs 1 no ...,nosuid
```

\_\_\_\_\_ Mount all file systems (including ROOT, excluding the read-only /usr) with the "logging" attribute by editing /etc/vfstab:

```
/dev/dsk/c0t0d0s0 /dev/rdisk/c0t0d0s0 / ufs 1 no remount,logging
/dev/dsk/c0t0d0s4 /dev/rdisk/c0t0d0s4 /usr ufs 1 no ro
/dev/dsk/c0t0d0s5 /dev/rdisk/c0t0d0s5 /var ufs 1 yes nosuid,logging
/dev/dsk/c0t0d0s6 /dev/rdisk/c0t0d0s6 /local ufs 2 yes nosuid,logging
swap - /tmp tmpfs - yes nosuid
/dev/dsk/c0t6d0s0 - /mnt/cdrom hsfs - no ro,nosuid
/dev/floppy - /mnt/floppy fd - no nosuid
```

[Previous](#) [Index](#) [Top](#)

## [Account Administration](#)

\_\_\_\_\_ Add in one or more administrative users and set secure passwords.

\_\_\_\_\_ Set the default SUPATH for ROOT and su in /etc/default/login: and /etc/default/su

```
SUPATH=/usr/sbin:/usr/bin:/usr/local/sbin:/usr/local/bin
```

\_\_\_\_\_ Set the default MANPATH in /etc/profile:

```
echo 'MANPATH=/usr/man:/usr/local/man;export MANPATH'>>\
/etc/profile
```

\_\_\_\_\_ Build the index so that "man -k" and "apropos" will work:

```
catman -w
```

\_\_\_\_\_ Make sure that /etc/default/login has the following lines (the default in Solaris 8):

```
# If CONSOLE is set, root can only login on that device.
# Comment this line out to allow remote login by root.
#
CONSOLE=/dev/console

# PASSREQ determines if login requires a password.
#
```

PASSREQ=YES

\_\_\_\_\_ Change the PASSLENGTH variable from 6 to 8 in /etc/default/passwd:

PASSLENGTH=8

\_\_\_\_\_ Touch the sulog to begin recording attempts by users to become ROOT:

```
touch /var/adm/sulog
```

\_\_\_\_\_ Add an entry into /etc/syslog.conf to record login failures:

```
auth.info                                /var/log/authlog
```

\_\_\_\_\_ Touch the authlog to begin recording authentication information:

```
touch /var/log/authlog
```

\_\_\_\_\_ Tell syslogd to re-read the control file:

```
kill -HUP `ps -ef | grep '/syslogd$' | \
grep -v grep | awk '{print $2}'`
```

\_\_\_\_\_ Lockout all accounts that do not require active logins (bin, adm, sys, lp, uucp, nobody, nobody4) and make the shell /dev/null in case the account is re-enabled somehow later [\[16\]](#).

```
for user in daemon bin adm sys lp uucp nuucp listen \
nobody noaccess nobody4
do
    /bin/passwd -l $user
    /usr/sbin/passmgmt -m -s /dev/null $user
done
```

\_\_\_\_\_ Make sure all users are in /etc/ftpusers to disable FTP access ([more...](#)):

```
/bin/sed 's/:.*//' < /etc/passwd >/etc/ftpusers
```

\_\_\_\_\_ Note any legitimate users of ftp and delete them from /etc/ftpusers:

---

\_\_\_\_\_ Don't allow any users other than ROOT to use the at command for scheduled command execution [\[17\]](#):

```
cd /etc/cron.d
rm -f cron.deny at.deny
echo root > cron.allow
echo root > at.allow
chown root:root cron.allow at.allow
chmod 400 cron.allow at.allow
```

\_\_\_\_\_ Remove the .rhosts file authentication entries from the /etc/pam.conf. These are identified by the library name pam\_rhosts\_auth.so.1 [\[19\]](#):

```
cp /etc/pam.conf /etc/pam.conf.rls
grep -v pam_rhosts_auth /etc/pam.conf.rls > /etc/pam.conf
```



```
rm -f /etc/pam.conf.rls
chown root:sys /etc/pam.conf
chmod 644 /etc/pam.conf
```

\_\_\_\_\_ Empty files should be supplied for /.rhosts, /.shosts, /.netrc, /etc/hosts.equiv, /etc/auto\_conf, /etc/auto\_home, /etc/dfs/dfstab and /etc/dfs/sharetab. These files can also be monitored by Tripwire to be sure that they stay empty [\[20\]](#).

```
#Run this as ROOT:
for file in /.rhosts /.shosts /.netrc /etc/hosts.equiv \
    /etc/auto_conf /etc/auto_home /etc/dfs/dfstab /etc/dfs/sharetab
do
    rm -f $file
    cat < /dev/null >$file
    chmod 000 $file
done
```

\_\_\_\_\_ Provide appropriate /etc/issue, /etc/issue.net and /etc/motd files with a warning that this is a proprietary system [\[21\]](#):

```
for file in issue issue.net motd
do
    rm -f /etc/$file
    ln -s /etc/banners/prototype /etc/$file
done
```

\_\_\_\_\_ Add banners for telnetd, ftpd and the eeprom to provide brief warnings about the use of the system:

```
echo 'BANNER="JFY Authorized users only. All accesses logged.\\n"' >>
/etc/default/telnetd
```

```
echo 'umask 022\\nBANNER="JFY Authorized users only. All accesses
logged."' >> /etc/default/ftpd
```

```
eeprom oem-banner="JFY Authorized users only. All accesses logged."
eeprom oem-banner\?=true
```

\_\_\_\_\_ Check and change if necessary the following entries in /usr/local/etc/sshd\_config:

```
PermitRootLogin no
X11Forwarding no
```

[Previous](#) [Index](#) [Top](#)

## [Solaris Auditing Features](#)

Solaris Auditing (SunSHIELD) is described in detail in the documentation <http://docs.sun.com/ab2/coll.47.8/SHIELD>. The full configuration is beyond the scope of this document, but basic configuration steps are given below. The subsystem can produce a huge amount of data, so be careful [\[22\]](#)!

\_\_\_\_\_ Enable the Solaris Auditing features:

```
echo y | /etc/security/bsmconv
```

\_\_\_\_\_ Create the /etc/security/audit\_control file:

```
dir:/var/audit
flags:lo,ad,-all,^-fm
naflags:lo,ad
minfree:20
```

\_\_\_\_\_ Activate the subsystem via crontab -e to run every hour (collection will not begin until the system is rebooted-- do not reboot yet):

```
0 * * * * /usr/sbin/audit -n
```

[Previous](#) [Index](#) [Top](#)

## Miscellaneous Security Enhancements

\_\_\_\_\_ Create an /etc/init.d/newnetconfig script file to setup more secure parameters for the ARP, IP and TCP protocols [\[23\]](#):

```
cat <<!EOF! >/etc/init.d/newnetconfig
nnd -set /dev/tcp tcp_conn_req_max_q0 10240
nnd -set /dev/tcp tcp_ip_abort_cinterval 60000
nnd -set /dev/ip ip_respond_to_timestamp 0
nnd -set /dev/ip ip_respond_to_timestamp_broadcast 0
nnd -set /dev/ip ip_respond_to_address_mask_broadcast 0
nnd -set /dev/ip ip_ignore_redirect 1
nnd -set /dev/ip ip_send_redirects 0
nnd -set /dev/ip ip_forward_src_routed 0
nnd -set /dev/ip ip_forward_directed_broadcasts 0
nnd -set /dev/ip ip_forwarding 0
nnd -set /dev/ip ip_strict_dst_multihoming 1
nnd -set /dev/arp arp_cleanup_interval 60000
nnd -set /dev/ip ip_ire_arp_interval 60000
nnd -set /dev/ip ip_respond_to_echo_broadcast 0
!EOF!
chmod 400 /etc/init.d/newnetconfig
ln -s /etc/init.d/newnetconfig /etc/rc2.d/S69newnetconfig
ls -lL /etc/init.d/newnetconfig /etc/rc2.d/S69newnetconfig
```

\_\_\_\_\_ Change the parameter TCP\_STRONG\_ISS=2 in /etc/default/inetinit [\[24\]](#).

```
#!/sbin/sh
ed - /etc/default/inetinit <<!EOF!
/^TCP_STRONG_ISS=1$/ s/=1$/=2/
w
w
q
!EOF!
```

\_\_\_\_\_ Prevent the use of STOP-A from the keyboard (also from unplugging the keyboard cable) by setting KEYBOARD\_ABORT=disable in /etc/default/kbd [\[24\]](#):

```
#!/sbin/sh
```

```
ed - /etc/disable/kbd << !EOF!  
/^#KEYBOARD_ABORT=default/ s/^#//  
w  
w  
q  
!EOF!
```

\_\_\_\_\_ Disable the "Service Access Controller" which is only used to enable modem logins on ttya and ttyb (This does NOT affect "headless system" logins where ttya is used for the console) [\[24\]](#):

```
#!/sbin/sh  
ed - /etc/inittab <<!EOF!  
/\sac / d  
w  
w  
q  
!EOF!
```

\_\_\_\_\_ Run the fix-mode script which was built on the development system [\[24\]](#):

```
mkdir -p /var/lp/logs  
/tmp/mnt/fix_modes/fix_modes      ### Errors from chmod are OK
```

\_\_\_\_\_ If so desired, remove packages that are no longer required (removing these makes patch administration more difficult, but also removes the sys-unconfig command):

```
pkgrm SUNWadmfw SUNWadmap SUNWadmc
```

\_\_\_\_\_ Create the initial Tripwire database and install it.

\_\_\_\_\_ Backup the Tripwire configuration file and database to CDROM and keep it available in the CDROM drive for reference.

[Previous](#) [Index](#) [Top](#)

## [Openboot Security](#)

\_\_\_\_\_ To enable the Sparc eeprom security feature [\[25\]](#) ([more...](#)):

```
eeprom security-mode=command  
<<< Supply new password twice as requested >>>  
eeprom security-#badlogins=0
```

\_\_\_\_\_ Check the auto-boot and diagnostic boot flags [\[26\]](#):

```
eeprom auto-boot?=true  
eeprom diag-device=disk  
eeprom diag-switch?=true
```

[Previous](#) [Index](#) [Top](#)

## [Testing](#)

\_\_\_\_\_ Reboot the system and either type STOP-A or unplug the keyboard. Verify that no openboot prompt appears.

\_\_\_\_\_ Reboot the system and power it off while it is coming back up several times to verify that the logging file system is working. There should no longer be any reason for the system to require a manual fsck.

\_\_\_\_\_ Check the files in /etc/rc?.d, they should look something like this:

```
# ls -Ra /etc/rc[23].d
/etc/rc2.d:
.          .NOS76nsd          S50newdevfsadm
..         .NOS80PRESERVE    S69inet
.NOS30sysid.net      .NOS88sendmail      S69netconfig
.NOS71ldap.client    .NOS93cacheos.finish S72local_sshd
.NOS71rpc            K28nfs.server       S72newinetsvc
.NOS71sysid.sys      README              S74newsyslog
.NOS72autoinstall    S01MOUNTFSYS        S74xntpd
.NOS73cachefs.daemon S05RMTMPFILES        S75cron
.NOS73nfs.client     S20syssetup         S75savecore
.NOS74autofs         S21perf             S88utmpd
.NOS74syslog         S99audit            S99audit

/etc/rc3.d:
.  .. README
#
```

\_\_\_\_\_ Make sure sendmail is not actively listening on port 25:

```
# telnet localhost 25
# telnet localhost 25
Trying ::1...
telnet: connect to address ::1: Connection refused
Trying 127.0.0.1...
telnet: Unable to connect to remote host: Connection refused
#
```

\_\_\_\_\_ Run an nmap scan to check for all open ports on the system:

```
# /usr/local/bin/nmap -sS -O 172.25.136.111
```

```
Starting nmap V. 2.54BETA22 ( www.insecure.org/nmap/ )
Interesting ports on grommet.iex.com (172.25.136.111):
(The 1541 ports scanned but not shown below are in state: closed)
Port State Service
22/tcp open  ssh
```

```
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=2.54BETA22%P=i686-pc-linux-gnu%D=5/8%Time=3AF83044%O=22%C=1)
TSeq(Class=TR%IPID=I%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%RID=E%RIPCK=E%UCK=E%ULEN=134%DAT=E)
```

Uptime 0.922 days (since Mon May 7 14:35:18 2001)

Nmap run completed -- 1 IP address (1 host up) scanned in 15 seconds

\_\_\_\_ Check to see which processes are running:

```
# ps -ef
UID PID PPID C STIME TTY TIME CMD
root 0 0 0 09:53:59 ? 0:01 sched
root 1 0 0 09:53:59 ? 0:00 /etc/init -
root 2 0 0 09:53:59 ? 0:00 pageout
root 3 0 0 09:53:59 ? 0:06 fsflush
root 747 1 0 11:17:59 console 0:00 -sh
root 880 747 0 15:11:23 console 0:00 ps -ef
root 174 1 0 09:54:12 ? 0:01 /usr/local/sbin/sshd
root 187 1 0 09:54:12 ? 0:00 /usr/sbin/syslogd -t
root 203 1 0 09:54:13 ? 0:00 /usr/lib/utmpd
root 191 1 0 09:54:13 ? 0:00 /usr/sbin/cron
#
```

\_\_\_\_ No in.routed, devfsadm or devfseventd processes are running.

\_\_\_\_ syslogd has the -t flag.

\_\_\_\_ No in.ndpd process is running (unless IPV6 is required).\

\_\_\_\_ Use the tcpdchk and tcpdmatch functions to test the tcp wrappers installation:

```
# ./tcpdmatch in.ftpd 172.16.34.18
client: address 172.16.34.18
server: process in.ftpd
matched: /etc/hosts.allow line 4
command: banners /etc/banners : ALLOW
access: granted
# ./tcpdmatch in.ftpd 172.16.30.8
client: address 172.16.30.8
server: process in.ftpd
matched: /etc/hosts.allow line 5
# ./tcpdmatch ssh 172.16.30.8
warning: ssh: no such process name in /etc/inet/inetd.conf
client: address 172.16.30.8
server: process ssh
matched: /etc/hosts.allow line 5
command: spawn (/usr/sbin/safe_finger -l @172.16.30.8 | /bin/mail -s
"Port Denial noted ssh-172.16.30.8" root) & : DENY
access: granted
# ./tcpdmatch ssh 172.16.4.10
warning: ssh: no such process name in /etc/inet/inetd.conf
```

```
client: address 172.16.4.10
server: process ssh
matched: /etc/hosts.allow line 1
command: banners /etc/banners : ALLOW
access: granted
# ./tcpdchk
warning: /etc/hosts.allow, line 1: ssh: no such process name in
/etc/inet/inetd.conf
#
```

\_\_\_\_\_ Use ssh to login and verify that no X11 forwarding is available:

```
$ ssh 172.16.6.111
xyzzzy@172.16.6.111's password:
Warning: Remote host denied X11 forwarding.
Last login: Thu May 3 15:38:38 2001 from 172.16.8.13
```

\_\_\_\_\_ Attempt ROOT access via ssh and verify that it is denied:

```
$ ssh root@172.16.6.111
root@172.16.6.111's password:
Permission denied, please try again.
```

\_\_\_\_\_ Verify the file system characteristics using mount:

```
# mount
/ on /dev/dsk/c0t0d0s0 read/write/setuid/intr/largefiles/logging/onerror=panic/dev=800000 on Thu May 3 15:47:56 2001
/usr on /dev/dsk/c0t0d0s6 read_only/setuid/intr/largefiles/onerror=panic/dev=800006 on Thu May 3 15:47:54 2001
/proc on /proc read/write/setuid/dev=3680000 on Thu May 3 15:47:55 2001
/dev/fd on fd read/write/setuid/dev=3740000 on Thu May 3 15:47:56 2001
/etc/mnttab on mnttab read/write/setuid/dev=37c0000 on Thu May 3 15:47:57 2001
/var on /dev/dsk/c0t0d0s1 read/write/nosuid/intr/largefiles/logging/onerror=panic/dev=800001 on Thu May 3 15:47:57 2001
/var/run on swap read/write/setuid/dev=1 on Thu May 3 15:47:57 2001
/tmp on swap read/write/nosuid/dev=2 on Thu May 3 15:47:59 2001
/local on /dev/dsk/c0t0d0s7 read/write/nosuid/intr/largefiles/logging/onerror=panic/dev=800007 on Thu May 3 15:47:59 2001
#
```

\_\_\_\_\_ Check the network paramters set using ndd in /etc/rc2.d/S69newnetconfig:

```
# ndd -get /dev/tcp tcp_strong_iss
2
# ndd -get /dev/ip ip_forwarding
0
<<< etc. >>>
```

\_\_\_\_\_ Attempt to su daemon and verify that no access is available:

```
# su - daemon
```

\_\_\_\_\_ Use su - root as an ordinary user and verify that the /var/log/authlog entries and /var/adm/sulog entries were made.

\_\_\_\_\_ As an ordinary user, verify that you cannot use the "at" command or define a crontab entry:

```
$ crontab -e
crontab: you are not authorized to use cron. Sorry.
$ at now
at: you are not authorized to use at. Sorry.
```

\_\_\_\_\_ Insert a CDROM into the drive and verify that it does not automount.

[Previous](#) [Index](#) [Top](#)

## [System Backups](#)

\_\_\_\_\_ Use the "ufsdump" command to take backups of all file systems to a removable storage device:

```
mt -f /dev/rmt/0 status    ### Rewind the tape
for fs in 0 1 6 7
do
    ufsdump 0uf /dev/rmt/0n /dev/rdisk/c0t0d0s${fs}
done
mt -f /dev/rmt/0 status    ### Rewind the tape
```

\_\_\_\_\_ Use the "ufsrestore" command to verify the backups of all file systems to another preformatted disk drive:

```
mt -f /dev/rmt/0 status    ### Rewind the tape
for fs in 0 1 6 7
do
    ufsrestore rf /dev/rmt/0n /dev/rdisk/c0t1d0s${fs}
done
mt -f /dev/rmt/0 status    ### Rewind the tape
```

\_\_\_\_\_ Optionally, copy the entire system to another system and make a CD image of all files-- the installation is small enough to easily fit on a single CDROM before other application software is loaded.

[Previous](#) [Index](#) [Top](#)

## [Physical Security](#)

1. Make sure that the system is in a locked room; make sure that either the walls are secure above ceiling tiles or that the surrounding rooms are also secure.
2. Use the locking hardware on the case so that the case cannot be opened and the eeprom cannot be swapped out by an attacker.
3. When typing root passwords, make sure that no one is "shoulder surfing." Likewise, in a secure environment make sure you're not in a position to be accused of "shoulder surfing." It's better if passwords are not written down at all, but if they must be then use a memorized four-digit "pin" so that the entire password need not be in writing.
4. Make sure that no one has left notes on the display or keyboard having password information.
5. Allow only trusted people to be alone in the room.
6. Consider the use of surveillance cameras in the room.

[Previous](#) [Index](#) [Top](#)

# **FAQ**

## Answers to Frequently Asked Questions

### **Why is the system not initially connected to a network?**

Attackers can use even the brief window of time when the system is first brought up to attack, since there are many services running and many un-patched vulnerabilities.

### **Why choose a minimal installation?**

The fewer OS components that are installed, the easier it is to protect the system from attackers. The components associated with Open Desktop have had many vulnerabilities [27], [28].

### **Why not use DHCP?**

A system to be used in the DMZ does not want to depend upon getting any information via the network, since such information could be spoofed by an attacker. DHCP can be used to supply vital information such as the IP address, the network mask, the default route and name servers and this information needs to be tightly controlled even if it is less convenient.

### **Why install NTP?**

Having the exact system time can be very important in coordinating log information between systems. The NTP package can ensure that system times can be coordinated. See <http://www.eecis.udel.edu/~ntp/> for more information.

### **Why install system documentation?**

It can be very useful to have system documentation available, especially in a multi-system environment where different versions of Solaris may be installed. Of course, it does take disk space, but with 18 GB disks being the default these days, this is insignificant. The latest SUN documentation can also be found online at <http://docs.sun.com>.

### **Why not review other packages that are installed?**

It is essential to review any other packages that are installed so as not to introduce additional vulnerabilities.

### **Why disable all ftp and telnet access?**

Both the FTP and Telnet protocols open up the system to password sniffing and are inherently insecure. The `/etc/ftpusers` file should be defined with all system level users in it just in case someone enables ftp access later.

### **Why install patches with the -nosave option?**

Using this option saves lots of disk space. If the patches are backed out, they must be backed out by hand in the reverse order of installation. Even SUN recommends against attempting to back out an entire cluster!

### **Why enable eeprom security?**

Physical access to the system would normally allow an attacker the option of gaining root access by booting any Solaris installation CDROM in single user mode, mounting the real root file system and temporarily eliminating the ROOT password or installing any desired trojans for later use. The OpenBoot Prom allows for additional physical security so that even if an attacker gains physical access considerable work must be done before the system can be booted from CDROM. The password used should NOT be the same as the system ROOT



password. It should be recorded in a separate, physically secure place since once it is set it cannot be cleared without replacing the SUN eeprom. Note also that a sufficiently experienced and determined attacker can still temporarily replace the SUN EEPROM, this just makes it a little harder! The parameter "security-#badlogins" can be reset to zero. Unsuccessful eeprom password attempts will be recorded.

### [Secure passwords](#)

Secure passwords are hard to pick! Typically secure passwords are made up of non-mnemonic symbols, numbers and upper-lower case letters (i.e. non-dictionary words). The complete password should never be written down unless it is placed inside a physically secure place like a locked safe. It should be known only to those people with a "need to know."

## [Previous](#) [Index](#) [Top](#)

## [References](#)

[1] [Pomeranz, H. Solaris Security Step by Step, Version 2.0, City, State: The SANS Institute, 2001. Introduction.](#)

[2] Ibid., p. 1

[3] Chouanard, J., [YASSP, \(Yet Another Solaris Security Package\).](#) 2001.

<http://www.yassp.org>

[4] Pomeranz, H., Ibid., p. 1.

[5] Pomeranz, H., Ibid., p. 6.

[6] [Seán, B., IT Security Cookbook, 2001.](#)

[http://www.boran.com/security/sp/Solaris\\_hardening3.html](http://www.boran.com/security/sp/Solaris_hardening3.html)

[7] Pomeranz, H., Ibid., p. 9.

[8] Pomeranz, H., Ibid., p. 8-9.

[9] Pomeranz, H., Ibid., p. 1.

[10] Pomeranz, H., Ibid., p. 3.

[11] Pomeranz, H., Ibid., p. 3.

[12] Pomeranz, H., Ibid., p. 22.

[13] Pomeranz, H., Ibid., p. 23.

[14] Pomeranz, H., Ibid., p. 24.

[15] Pomeranz, H., Ibid., p. 12.

[16] Pomeranz, H., Ibid., p. 17.

[17] Pomeranz, H., Ibid., p. 18.

[18] Pomeranz, H., Ibid., p. 18.

- [19] Pomeranz, H., Ibid., p. 17.
- [20] Pomeranz, H., Ibid., p. 17.
- [21] Pomeranz, H., Ibid., p. 18.
- [22] Pomeranz, H., Ibid., p. 15-16.
- [23] Pomeranz, H., Ibid., p. 10.
- [24] Pomeranz, H., Ibid., p. 21.
- [25] Pomeranz, H., Ibid., p. 19,20.
- [26] [Campioni, J., \*Solaris 8 Installation Checklist\*, SANS Institute, 2000, Bethesda, MD, p. 1](#)
- [27] [Scambray, J. McClure, S. and Kurtz, G., \*Hacking Exposed\*, Berkely, California: Osborne/McGraw-Hill, 2000.](#)  
<http://www.hackingexposed.com>
- [28] [Security Focus. 2001. \*Security Focus web site\*.](#)  
<http://www.securityfocus.com>

[Solaris 2 FAQ, 2001.](#)

[http://www.science.uva.nl/pub/solaris/solaris2/.](http://www.science.uva.nl/pub/solaris/solaris2/)

[Solaris 2 Security FAQ. 2001.](#)

<http://www.itworld.com/Comp/2377/security-faq/>

Solaris 2 Tuning your TCP/IP Stack, and more., 2001,

<http://www.sean.de/Solaris/tune.html>

[Solaris FAQ, 2001,](#)

<http://www.faqs.org/faqs/Solaris2/>

[Previous](#)   [Index](#)   [Top](#)

## [Appendix INET](#)

Quoted from Solaris Security, Appendix A [Pomeranz,01]:

### **REPLACEMENT /ETC/INIT.D/NEWINETSVC SCRIPT**

```
#!/sbin/sh
/usr/sbin/ifconfig -au netmask + broadcast +
if [ -f /usr/sbin/in.named -a -f /etc/named.conf ]; then
/usr/sbin/in.named
echo "starting internet domain name server."
fi
```

```
#mcastif=$(uname -n)
#echo "Setting default interface for multicast: \c"
#/usr/sbin/route add -interface -netmask "240.0.0.0" "224.0.0.0" "$mcastif"
# Run inetd in "standalone" mode (-s flag)
#/usr/sbin/inetd -s -t
```

## REPLACEMENT /ETC/INIT.D/NEWINETSVC SCRIPT

[Previous](#) [Index](#) [Top](#)

---

**Author: Ben A. Laws, Jr.**

**For: JFY, Inc. Released to SANS for GCUX Exam.**

**Revised: May 08, 2001 .**

© SANS Institute 2000 - 2005, Author retains .