Interested in learning more about security?

SANS Institute
InfoSec Reading Room

This paper is from the SANS Institute Reading Room site. Reposting is not permitted without express written permission.

Configuration and Patch Verification on Solaris Systems

Solaris patch levels can be difficult to manage any time that more than a few systems are involved. Sun provides the PatchPro, Patch Manager and Patchdiag utilities, as well as providing the Explorer 1 script, but these are not as customizable as is sometimes required in a non-standard installation. In particular, these tools do not give a good sense whether or not there are security related patches that need to be installed. In this paper, I discuss two configuration monitoring scripts which I have used to ensure that...

Copyright SANS Institute
Author Retains Full Rights

AD
Symantec™
Endpoint Protection 12
The next generation of reputation-based security
Download the beta
Symantec.
Confidence in a connected world.

Configuration and Patch Verification on Solaris Systems

Scott Cromar
GSEC Assignment Version 1.4b-Option 1
22 Jan 2003

Abstract:
Solaris patch levels can be difficult to manage any time that more than a few systems are involved. Sun provides the PatchPro, Patch Manager and Patchdiag utilities, as well as providing the Explorer[1] script, but these are are not as customizable as is sometimes required in a non-standard installation. In particular, these tools do not give a good sense whether or not there are security-related patches that need to be installed.

In this paper, I discuss two configuration monitoring scripts which I have used to ensure that systems are up to patch and configuration standards. I have found these scripts is to be transparent enough to allow significant customization, but powerful enough to check for most key configuration issues.

Patching Philosophy:
In general, once a production system is working, it is left alone. There is often no point in adding patches for problems not experienced or which are deemed to be minor. Patches are usually only added to a production system as a result of troubleshooting a problem on a system or as a result of a reported security issue.

Production systems are, by their very nature, sensitive to issues of confidentiality, integrity, or availability. While it is inconvenient to make changes to critical production systems, it is important to keep them up-to-date on security patches.

There are important issues that may be solved by a patch not normally considered a "security" patch. Regardless of whether these are listed in Sun's list of "Security Patches," there must be a mechanism for adding these to the auditing script in order to fix issues affecting confidentiality, integrity or availability. Since some of these issues may be installation-specific, it is important that the auditing script be straightforward enough to allow customization for each environment.

In general, point patches should be applied to fix security issues as they arise, with one major caveat. A general patch update should be performed whenever a major security (or stability) issue is corrected by the relevant kernel, libc, or libthread patch. This general update would include the Sun recommended cluster as well as driver and software patches specific to the given system.

In practice, upgrading to the recommended cluster on a semi-regular basis means that systems are never too terribly far from the Sun recommended patches. Security issues involving kernel, libc and libthread patches seem to emerge every 90 days or so.

[1] These are available from http://sunsolve.sun.com, along with discussions of the capabilities of each.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Keeping relatively current on patches can be important when requesting support from Sun Service. If patches levels are very far behind, you are likely to be told to upgrade in order to receive adequate support.

(Sun performs the most rigorous testing on its Recommended Cluster as a group, so there is a benefit to upgrading the entire cluster at once in order to avoid conflicts. Unless your environment is large enough to support rigorous testing of your own patch cluster, I recommend sticking as closely as possible to the Sun Recommended Cluster.)

Obviously, the proposed patch set must be applied in the context of a structured change control procedure, including rigorous testing and a documented backout procedure. Sun recommends that patches be applied to at least a QA/Integration system prior to deployment to a main production system. Depending on the criticality of the system, they suggest that it may be desirable to apply patches one at a time.[2] If the patch is less critical, perhaps the patch can be rolled into the testing cycle for a larger change deployment.

While applying patches one at a time may be overkill, it is critical to employ an appropriate backout mechanism. patchrm is an appropriate mechanism where only a few patches are deployed, or where the server can be taken down for maintenance.

Where the system is clustered, it may be relatively easy to apply the patches to the production system. Patches are applied to one server, testing is completed, services are switched to that server, then the second server is upgraded.

Alternatively, a backup can be made to tape for a restore in the event of a backout. In my experience, this is usually even slower than patchrm. Backups can also be made to disk using dd or ufsdump/ufsrestore followed by an installboot command. Make sure that you have a well-documented and tested procedure for a "bare-metal" restore of the system from your backup medium.

(Regardless of the mechanism, a full backup should be taken prior to a patch installation. In my opinion, however, backups should only be used as a backout mechanism as a last resort.)

In the case of a truly critical single server where minimal downtime is absolutely required, it might make sense to use Solaris Live Upgrade as the backout mechanism. This can be used to create a replica of the boot image. The patches can be applied to the copy, and the copy can even be moved to another system for testing. Finally, Solaris Live Upgrade can be used to switch to the copy as the live production environment. The time to perform this switch is roughly the same as the amount of time required for a reboot.

Script History:

Over the last few years, I have discovered that I need to have a relatively quick, easily automated way to verify that the Solaris systems I administer are adequately patched for security issues. We can use a tool such as Patchdiag to verify whether the system is at the current patch level for Sun's

[2] Radhakrishnan, Ramesh. "A Patch Management Strategy for the Solaris Operating Environment." Sun Microsystems. Jan 2003. URL: http://www.sun.com/solutions/blueprints/0103/817-1115.pdf.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Recommended and Security patch clusters. But what is really needed, especially for critical systems, is some sense of how important each of these patches is. If a patch can be avoided by turning off an unneeded service (or verifying that the affected service is already turned off), this is usually preferable to rushing a patch installation.

In 1999, while I was the Solaris administrator for CIT at Princeton University, Bill Bridges and I created a "check.advisories" script. check.advisories compared a system's configuration against a list of patches, which we generated from a number of sources. This script has since evolved into the "check-advisories.sh" script listed below.

In addition, I created a host-audit.sh script to look at configuration vulnerabilities that were not correctable by patches.

Requirements for check-advisories.sh:
The check-advisories.sh script had a number of requirements:

1) The script itself needed to be in Bourne shell to guarantee that it would run across our environment. (At the time we first wrote the script, we were supporting SunOS 4.1.4, in addition to Solaris 2.5.1, 2.6, 7 and 8. Perl and ksh were not universally available on our hosts.)
2) The patches to be examined should be listed in relatively concise configuration files. Each record in the configuration file should ideally fit on a single line of the file. Vulnerability descriptions, in particular, needed to be concise enough to fit on a single line, but point to other sources where more detail was available.
3) The entries for each patch needed to include a meaningful name for the service being patched, a patch number, a minimal patch level, and references about the vulnerabilities that were patched.
4) There needed to be some mechanism for determining whether the patch was appropriate for a given system. For example, it made no sense to report that CDE patches were not installed if the CDE packages themselves were not installed.
5) We needed to be able to bring in information from several different sources. It had long since become obvious that Sun's recommended and security patch lists were inadequate and lacked detail, and that the Bugtraq archives were not comprehensive enough for our needs. We obviously needed to include information about CERT advisories and Sun Security Bulletins as well.
6) We needed to be able to list multiple vulnerabilities per patch number. This is important because sometimes one patch level fixes a critical buffer overflow that is actively being exploited, and a later patch level fixes something less ominous like a memory leak that is potentially exploitable to create a denial of service attack.
7) We needed to be able to list multiple bugs per patch level, since one patch will frequently fix several vulnerabilities.

The script has continued to evolve to fit changes in the Solaris landscape. It no longer supports SunOS 4.1.4 or Solaris 2.5.1 or 7, since none of these are in wide use any more. Current

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

vulnerability files for Jan 21, 2003 are included as appendices below. These are updated manually based on traffic from the Bugtraq[3] lists as well as changes on the biweekly Solaris patch reports[4].

The format of the vulnerability configuration file has changed as well. For each supported version of Solaris, a file /usr/local/etc/vuln-list-5.x is created. Each record in these files is a colon-separated list of the following fields:

1) Name of item being patched.
2) File replaced by the patch. (Preferably the specific file that has the identified vulnerability.)
3) Patch number.
4) Minimum patch version number.
5) Brief references to sources of information about the bugs covered by the patch.

The reference field (#5 above) is not intended to be a complete list of the bugs covered by the patch. It is a list of references to bug descriptions that can be used to describe why the patch is important. The most frequent use of this information is to allow the system administrator to gauge the importance of applying the patch now and to communicate this assessment to the business users of the system.

In general, I list these references in order of importance/criticality. This is clearly a subjective measurement, but it has evolved as the most useful way to list them.

The references are coded as follows:

1) SB: Sun Bug number. One-line descriptions of these can be obtained from the patch description of the patch in question at http://sunsolve.sun.com. More complete descriptions for some bugs can be obtained from the Sunsolve site. Other bugs reports are not available without a service contract.
2) SSB: Sun Security Bulletin. These are listed at http://sunsolve.sun.com/pub-cgi/secBulletin.pl. These are generally very detailed, but they cover only a small proportion of critical security issues.
3) SFBID: Security Focus (Bugtraq) Bug ID. These can be looked up at http://www.securityfocus.com. These generally include significant information about the vulnerability itself, but are frequently outdated when it comes to how to fix the vulnerability.
4) CA: CERT Advisory. These can be found at http://www.cert.org/nav/index_red.html. Due to name recognition, I find that business users of systems are more likely to give permission for patch upgrades when there is a CERT Advisory number that can be referenced.

I have toyed with the idea of creating longer bug descriptions, similar to those found in tools such as Nessus. (The original 1999 script pointed to a web page containing information about each

[3] Bugtraq archives are viewable at http://www.securityfocus.com. Free email subscriptions to Bugtraq are also available on SecurityFocus.

[4] Solaris patch reports are updated twice per month, usually at the beginning and middle of each month. They can be viewed at http://sunsolve.sun.com. Service contract holders can request email updates of these reports as they become available.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

vulnerability as well as links to the original advisories.)[5] In the end, I have found that having the reference to a more complete data source is more useful, as well as allowing for much more concise reports.

check-advisories.sh Listing:

```
#!/bin/sh
# Script to automate checking of a Solaris 2.6-9 system for compliance
# with several security advisories.
#
# Scott Cromar and Bill Bridges
# 15 Apr 1999
#
# Copyright 1999,2000,2001,2002,2003 Trustees of Princeton University
#
# This script may only be copied or run for educational purposes or by
# persons authorized to examine the security of the systems on which
# it is run.  Any other uses of this script are a violation of Princeton
# University's copyright.
#
# Current version: 21 Jan 2003
# Driver script to check a system against a vulnerability database

OS=`uname -sr | awk '{print $2}'`
VULNDB=/usr/local/etc/vuln-list-${OS}
EMAIL=root
TEMPFILE=/var/adm/auditadvtemp.$$
REPORT=/var/adm/auditadvreport.$$
SHOWREV=/var/adm/showrev.$$

# Initialize working files
rm -rf ${SHOWREV} ${TEMPFILE} ${REPORT}
touch ${REPORT}
/bin/showrev -p | cut -c1-30 > ${SHOWREV}

# Read, parse and process each line
cat ${VULNDB} | while read LINE
do
  VULNNAME=`echo ${LINE} | awk -F: '{print $1}'`
  VULNFILE=`echo ${LINE} | awk -F: '{print $2}'`
  VULNPATCHNO=`echo ${LINE} | awk -F: '{print $3}'`
  VULNPATCHVER=`echo ${LINE} | awk -F: '{print $4}'`
  VULNBUGID=`echo ${LINE} | awk -F: '{print $5}'`
```

[5] Cromar, Scott. "Solaris Security Advisories." Princeton University. May 2000. URL: http://www.princeton.edu/~cromar/SolarisSecurity/. Note that this page is hopelessly out of date. Since I no longer work for Princeton, I no longer have access to the page to update it. This link is included as a historical reference.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

```
  if [ -f ${VULNFILE} ]; then
    VERS=`cat ${SHOWREV} | awk '{print $2}' | grep ${VULNPATCHNO} | sort -rn |
 head -n 1 | sed -e 's/^.....-//'`
    if [ 0${VERS} -lt ${VULNPATCHVER} ]; then
      echo "${VULNPATCHNO}-${VULNPATCHVER} required for ${VULNBUGID} hole in
${VULNNAME}" >> ${REPORT}
    fi
  else
    echo "${VULNFILE} not present"
    exit 1
  fi
done

cat ${REPORT}

# Cleanup
rm -f ${SHOWREV} ${TEMPFILE} ${REPORT}
```

© SANS Institute 2003, Author retains full rights

Requirements for host-audit.sh:
System configuration is even more important than OS patch level. Unused services should be turned off. Where they cannot be turned off, access should be restricted as far as possible. Files and directories should have the minimal permissions necessary to allow proper system functioning.

check-advisories.sh was very useful for checking patch levels, but we needed to go further in evaluating our systems with respect to our established standards. We evaluated several key issues in our host-audit.sh script. We selected those issues that we identified as being the most critical and the most likely to be misconfigured in our environment.

check-advisories.sh is called by host-audit.sh to provide a list of patch issues with the system.

The list of settings checked by host-audit.sh is by no means a comprehensive list of proper security settings. A number of these issues are covered now covered by the CIS Benchmark script.

The CIS Benchmark is a tool for checking a system against the CIS recommendations for "the prudent level of minimum due care for operating system security[6]." It is a significant aid in checking a number of configuration settings that are easily overlooked when setting up a Solaris system. I have been using it since it came out, and I have been very pleased with the level of coverage available from the CIS Benchmark scripts.

There is significant overlap of this script with the CIS Benchmark[7] recommendations for Solaris, but there are also distinct differences in coverage between the two. Following the host-audit.sh listing, I list the important items covered by the CIS Benchmark that are not covered by host-audit.sh.

One advantage that host-audit.sh has over its CIS counterpart is that it takes into account all security patches, not just those that are in the Recommended patch cluster. The CIS Benchmark's approach to patches is to check to see if a patch—any patch—has been installed recently. The assumption is that if one patch has been installed, other required patches have been updated as well.

Unfortunately, it is sometimes the case that a patch fails to install as part of the Recommended Cluster installation. Even when a patch is included in the Recommended Cluster, and a cluster installation is attempted, unless the logs are read assiduously, an important patch can slip through the cracks. More frequently, a patch that is not part of the Recommended Cluster can be overlooked when rolling a patch upgrade for a given server.

Obviously, the primary disadvantage of the host-audit.sh and check-advisories.sh approach is that the vuln-list files have to be maintained assiduously. Also, the messages provided by host-audit.sh are much more telegraphic than those provided by CIS for the same issues. This is by design, but it can be intimidating to someone who is not familiar with the log messages.

---

[6] Center for Internet Security. "Solaris Benchmark v 1.1.0." Oct 2002. URL: http://www.cisecurity.org/bench_solaris.html

[7] The CIS Benchmark scoring script includes substantial information about correcting insecure configurations. The document about secure configuration is at https://www.cisecurity.org/tools2/solaris/SolarisBenchmark.pdf

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

In general, host-audit.sh is also less comprehensive in coverage than is the CIS script. I run the CIS script on my servers in addition to host-audit.sh in order to check on configuration items covered by one, but not the other.

Many of the settings in this script were influenced by the Watson/Noordergraaf series of Sun blueprints[8].

We wanted host-audit.sh to check for the following configuration standards:

1) Check patch levels against security advisories. We implemented this by calling check-advisories.sh.
2) Check /etc/system for several key security settings:
    a. noexec_user_stack
    b. nfssrv:nfs_portmon
    c. Network interface autonegotiation turned off and proper speed/mode settings enabled.
3) Verify that the /etc/rc2.d/S69netconfig script was installed. A copy of the netconfig script is included as an appendix. It incorporates recommendations from Sun[9] on how to securely configure network access for Solaris, as well as comments from the "Tuning Your TCP/IP Stack" site[10]. (Note that the netconfig script will need to be altered for systems that act as firewalls, multihomed hosts or routers. Specific changes are in the above-referenced Sun white paper.)
4) Verify that /etc/rc2.d/S00umask was installed.
5) Check a configurable list of initialization scripts to make sure that unwanted services were not enabled.
6) Check that vold, if enabled, would mount floppies and CDs as nosuid devices.
7) Verify that system accounts are locked.
8) Verify that direct remote root logins were disabled.
9) Check inetd.conf:
    a. Desired services were protected by tcp wrappers, where appropriate.
    b. Other services were disabled.
10) The Xaccess permissions were as restrictive as possible.
11) /.rhosts and /etc/hosts.equiv were empty.
12) /etc/syslog.conf included mail.debug and auth.info logging levels.
13) Crontab ownerships and permissions were set appropriately.

The host-audit.sh script presented below includes most of the important pieces of the original, with several local customizations removed. I have also removed the previous version's method for checking file and directory permissions. Instead, I recommend running Sun's fix-modes[11] script

---

[8] Watson, Keith and Noordergraaf, Alex. "Solaris Operating Environment Security." Sun Microsystems. Dec 2002. URL: http://www.sun.com/solutions/blueprints/1202/816-5242.pdf.

[9] Watson, Keith and Noordergraaf, Alex. "Solaris Operating Environment Network Settings for Security." Sun Microsystems. Dec 2000. URL: http://www.sun.com/solutions/blueprints/1200/network-updt1.pdf.

[10] Vöckler, Jens S. "Solaris 2.x-Tuning your TCP/IP Stack." URL: . http://www.sean.de/Solaris/soltune.html.

[11] CIS makes this script available at ftp://ftp/CISecurity.org/pub/pkgs/Solaris/fix-modes.tar.Z

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

against each system. aset (which is included in the Solaris distribution) can also be used to set permissions on system files and directories, though my experience with this is that it tends to be too draconian.

host-audit.sh Listing:

```
#!/bin/sh
#
# Scott Cromar
#
# Copyright 1999,2000,2001,2002,2003 Trustees of Princeton University
#
# This script may only be copied or run for educational purposes or by
# persons authorized to examine the security of the systems on which
# it is run.  Any other uses of this script are a violation of Princeton
# University's copyright.
#
# Current version: 21 Jan 2003
# Script to audit a system with respect to Solaris
# System Standards.

# Initialize variables/files
REPORT=/var/adm/
IFTYPE=/var/adm/iftype

rm -rf ${REPORT} ${IFTYPE}
touch ${REPORT} ${IFTYPE}
chmod 600 ${REPORT} ${IFTYPE}

echo "HOST AUDIT for `/bin/hostname`" >> ${REPORT}
/bin/date >> ${REPORT}
echo >> ${REPORT}

echo "##############################################################">>
${REPORT}
echo >> ${REPORT}

echo "check-advisories.sh" >> ${REPORT}
echo >> ${REPORT}

# Call check-advisories.sh
if [ -x /usr/local/bin/check-advisories.sh ]; then
  /usr/local/bin/check-advisories.sh >> ${REPORT}
  echo >> ${REPORT}
  ls -l /usr/local/etc/vuln*`uname -sr | awk '{ print $2 }' | awk -F. '{ print $2 }'` | awk '{ print $6
" " $7 " " $8 " " $9 }' >> ${REPORT}
  echo "Verify that check-advisories vuln databases are up-to-date" >> ${REPORT}
else
```

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

```
   echo "Install check-advisories.sh and /usr/local/etc/vuln* files" >> ${REPORT}
fi
echo "##########################################################">>
{REPORT}
echo >> ${REPORT}

echo "/etc/system Checks" >> ${REPORT}

# Check /etc/system
SYSTEMCHECK=`grep "noexec_user_stack" /etc/system | grep 1 | grep -v \^'*' | wc -l`
if [ ${SYSTEMCHECK} -lt 2 ]; then
   echo >> ${REPORT}
   echo "Set noexec_user_stack in /etc/system" >> ${REPORT}
fi

NFSCHECK=`grep "nfssrv:nfs_portmon" /etc/system | grep -v \^'*' | wc -l`
if [ ${NFSCHECK} -lt 1 ]; then
   echo >> ${REPORT}
   echo "Set nfs_portmon in /etc/system" >> ${REPORT}
fi

# Check to verify that interface speed/mode nailed down
ls /etc/hostname* | awk -F. '{print $2}' > ${IFTYPE}
HMECHECK=`grep hme ${IFTYPE} | wc -l`
QFECHECK=`grep qfe ${IFTYPE} | wc -l`
ERICHECK=`grep eri ${IFTYPE} | wc -l`
if [ ${HMECHECK} -gt 0 ]; then
   HMESETCHECK=`grep hme_adv_autoneg_cap /etc/system | grep -v \^'*' | wc -l`
   if [ ${HMESETCHECK} -lt 1 ]; then
      HMENDDCHECK=`grep hme_adv_autoneg_cap /etc/rc2.d/S69sethme | grep -v ^# | wc
-l`
      if [ ${HMENDDCHECK} -lt 1 ]; then
         echo >> ${REPORT}
         echo "Nail down hme interfaces in /etc/system or /etc/rc2.d/S69sethme" >>
${REPORT}
      fi
   fi
fi

if [ ${QFECHECK} -gt 0 ]; then
   QFESETCHECK=`grep "qfe:adv_autoneg_cap" /etc/system | grep -v \^'*' | wc -l`
   if [ ${QFESETCHECK} -lt 1 ]; then
      QFENDDCHECK=`grep "qfe adv_autoneg_cap" /etc/rc2.d/S69sethme | grep -v ^# | wc
-l`
      if [ ${QFENDDCHECK} -lt 1 ]; then
         echo >> ${REPORT}
```

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

```
        echo "Nail down qfe interfaces in /etc/system or /etc/rc2.d/S69sethme" >>
${REPORT}
      fi
   fi
fi

if [ ${ERICHECK} -gt 0 ]; then
   ERISETCHECK=`grep "eri:adv_autoneg_cap" /etc/system | grep -v \^'*' | wc -l`
   if [ ${ERISETCHECK} -lt 1 ]; then
      ERINDDCHECK=`grep eri autoneg_cap /etc/rc2.d/S69sethme | grep -v ^# | wc -l`
      if [ ${ERINDDCHECK} -lt 1 ]; then
        echo >> ${REPORT}
        echo "Nail down eri interfaces in /etc/system or /etc/rc2.d/S69sethme" >>
${REPORT}
      fi
   fi
fi

echo >> ${REPORT}
echo "##########################################################">>
{REPORT}
echo >> ${REPORT}

echo "Check Startup Files" >> ${REPORT}
echo >> ${REPORT}

if [ ! -x /etc/rc2.d/S69netconfig ]; then
   echo >> ${REPORT}
   echo "Install /etc/rc2.d/S69netconfig" >> ${REPORT}
fi

if [ ! -x /etc/rc2.d/S00umask ]; then
   echo >> ${REPORT}
   echo "Install S00umask" >> ${REPORT}
fi

for RCSCRIPTS in sysid llc2 asppp ppp uucp ldap autoinstall slpd cachefs nfs.client autofs
xntpd PRESERVE power wbem cacheos ncalogd llim webastart dtlogin dhcp nfs.server
apache boot.server snmpdx dmi samba
do
   if [ -x /etc/rc[2-3].d/S[0-9][0-9]${RCSCRIPTS}* ]; then
      echo >> ${REPORT}
      echo "Disable ${RCSCRIPTS} in `ls /etc/rc[2-3].d/S[0-9][0-9]${RCSCRIPTS}*`
 if unnecessary" >> ${REPORT}
   fi
done
```

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

```
VOLMGTCHECK1=`grep -v \^\# /etc/rmmount.conf | grep mount | grep -i hsfs | grep
nosuid | wc -l`
VOLMGTCHECK2=`grep -v \^\# /etc/rmmount.conf | grep mount | grep -i ufs | grep n
osuid | wc -l`
if [ ${VOLMGTCHECK1} -lt 1 -o ${VOLMGTCHECK2} -lt 1 ]; then
   echo >> ${REPORT}
   echo "Set nosuid for hsfs and ufs in /etc/rmmount.conf" >> ${REPORT}
fi


echo >> ${REPORT}
echo "##############################################################">>
${REPORT}
echo >> ${REPORT}

echo "Check Accounts" >> ${REPORT}
echo >> ${REPORT}

echo "Check Accounts" >> ${REPORT}
echo >> ${REPORT}

# Make sure system accounts have locked passwords
for ACCOUNT in daemon bin sys adm lp uucp nuucp listen nobody noaccess nobody4
backupadmin
do
   PASSWDCHECK=`grep \^${ACCOUNT} /etc/shadow | awk -F: '{print $2}' | grep -v NP |
grep -v LK | wc -l`
   if [ ${PASSWDCHECK} -gt 0 ]; then
      echo >> ${REPORT}
      echo "Disable ${ACCOUNT} password" >> ${REPORT}
   fi
   FTPUSERCHECK=`grep \^${ACCOUNT} /etc/ftpusers | wc -l`
   if [ ${FTPUSERCHECK} -lt 1 ]; then
      echo >> ${REPORT}
      echo "Add ${ACCOUNT} to /etc/ftpusers" >> ${REPORT}
   fi
done

# Make sure root can only log in through console
CONSOLECHECK=`grep -v \^\# /etc/default/login | grep CONSOLE | wc -l`
if [ ${CONSOLECHECK} -lt 1 ]; then
   echo ${REPORT}
   echo "Disable remote root login in /etc/default/login" >> ${REPORT}
fi

echo >> ${REPORT}
```

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

```
echo "##############################################################">>
${REPORT}
echo >> ${REPORT}

echo "Check Network Services" >> ${REPORT}
echo >> ${REPORT}

for WRAPPED in ftp telnet shell login finger exec uucp tftp talk
do
  CHECKWRAPPED=`grep \^${WRAPPED} /etc/inet/inetd.conf | grep -v tcpd | wc -l`
  if [ ${CHECKWRAPPED} -gt 0 ]; then
    echo >> ${REPORT}
    echo "${WRAPPED} service must be wrapped if enabled" >> ${REPORT}
  fi
done

for ENABLED in fs shell login exec comsat uucp tftp systat netstat echo daytime chargen
100232 rquotad rusers spray walld rstatd rexd 100083 ufsd 10021 fs 100235 100134
printer 100234 100146 100147 100150 dtspc 100068 bootp rpc.metad rpc.metamhd
do
  CHECKENABLED=`grep \^${ENABLED} /etc/inet/inetd.conf | wc -l`
  if [ ${CHECKENABLED} -gt 0 ]; then
    echo >> ${REPORT}
    echo "${ENABLED} service should be turned off if unnecessary" >> ${REPORT}
  fi
done

XACCESS=`grep -v \^\# /usr/dt/config/Xaccess | grep [A-z] | grep -v localhost | wc -l`
if [ ${XACCESS} -gt 0 ]; then
  echo >> ${REPORT}
  echo "Xaccess granted to the following in /usr/dt/config/Xaccess:" >> ${REPORT}
  grep -v \^\# /usr/dt/config/Xaccess | grep [A-z] | grep -v localhost >> ${REPORT}
fi

if [ -s /.rhosts ]; then
  echo >> ${REPORT}
  echo "/.rhosts contains:" >> ${REPORT}
  cat /.rhosts >> ${REPORT}
fi

if [ -s /etc/hosts.equiv ]; then
  echo >> ${REPORT}
  echo "/etc/hosts.equiv contains:" >> ${REPORT}
  cat /etc/hosts.equiv >> ${REPORT}
fi
```

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

```
SYSLOGCHECK=`grep -v \^\# /etc/syslog.conf | grep messages | grep mail.debug | grep
auth.info| wc -l`
if [ ${SYSLOGCHECK} -lt 1 ]; then
   echo >> ${REPORT}
   echo "Turn on auth.info and mail.debug in /etc/syslog.conf" >> ${REPORT}
fi

echo >> ${REPORT}
echo "##############################################################">>
${REPORT}
echo >> ${REPORT}

echo "Permissions" >> ${REPORT}
echo >> ${REPORT}

CRONPERMS=`ls -l /var/spool/cron/crontabs/* | grep -v '\-r\-\-\-\-\-\-\-\-' | wc -l`
if [ ${CRONPERMS} -gt 0 ]; then
   echo >> ${REPORT}
   echo "fix crontab permissions" >> ${REPORT}
fi

CRONOWNER=`ls -l /var/spool/cron/crontabs/* | grep -v root | wc -l`
if [ ${CRONOWNER} -gt 0 ]; then
   echo >> ${REPORT}
   echo "fix crontab ownerships" >> ${REPORT}
fi

echo >> ${REPORT}
echo "##############################################################">>
${REPORT}
echo >> ${REPORT}

cat ${REPORT}

rm -f ${REPORT} ${IFTYPE}
```

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Limitations of host-audit.sh and check-advisories.sh:
The host-audit.sh and check-advisories.sh scripts were intended to reduce the amount of administrator time and effort required to verify compliance with security advisories and a few of our configuration standards. These scripts are not sufficient, in themselves, to guarantee a secure or even well-configured system.

Even if the configuration steps listed above are taken, there are still several issues that should be addressed in a secure, well-configured system. I strongly recommend running a broad variety of security tools against your systems. In particular, the CIS Benchmark is a very useful tool for pointing out several frequently-overlooked configuration issues.

Below I list several security issues which would be caught by the CIS Benchmark, but which are not checked by host-audit.sh and check-advisories.sh. This section is provided in order to help users of host-audit.sh create a reasonably secure configuration of their systems. I strongly recommend that the CIS Benchmark be used in conjunction with host-audit.sh, since they complement each other's weaknesses.

User Accounts:

- No attempt has been made to enforce even a rudimentary password policy in these scripts. A secure system will have a method for verifying that passwords are reasonably strong, and that they are changed on a regular basis. The /etc/shadow man page[12] includes information on how to set minimum days between password changes, the maximum number of days the password is valid, the number of warning days allowed, the number of inactive days allowed, and an absolute expiration date.

- Some method should also be employed to avoid weak passwords. npasswd[13] is a drop-in replacement for /bin/passwd which will perform basic weak password checking. Crack[14] and John the Ripper[15] are password crackers which may be used by a system administrator to check the strength of users' passwords.

  (Ideally, of course, no passwords would be used. Token-based authentication or another two-factor authentication scheme would be much more secure. As a practical matter, though, most authentication systems are still solely password-based.)

- The /etc/passwd, /etc/shadow and /etc/group files should be examined for "+" entries left over from previous incarnations, as well as inactive accounts and UID=0 accounts other than root.

- User home directories should have relatively restrictive permissions, especially when it comes to the "dot" configuration files. Users should not have their own .netrc or .rhosts files.

---

[12] http://docs.sun.com/db/doc/816-0219/6m6njqbc2?a=view#indexterm-170, or "man shadow" at a Solaris login prompt.
[13] http://www.utexas.edu/cc/unix/software/npasswd
[14] ftp://ftp.cerias.purdue.edu/pub/tools/unix/pwdutils/crack/
[15] http://www.openwall.com/john/

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Secure Connections and Connection Logging:

- Telnet and ftp are fundamentally insecure because they ship passwords and data across the network in the clear. Alternatives such as Kerberos[16] and Openssh[17] should be considered seriously in order to protect both passwords and data.

- While host-audit.sh does check that TCP Wrappers are installed on relevant services, it does not check that hosts.allow and hosts.deny are set up reasonably. These should be set to the most restrictive level feasible. Also, while host-audit.sh does verify that auth.info is logged to /var/adm/messages, it is advisable that the information be logged to a separate syslog server as well.

Sendmail status:

- /etc/init.d/sendmail is not examined to make sure that "mode" is set to "" and/or that /etc/rc2.d/S88sendmail is disabled. This is important for all systems that are not actually performing mail delivery services.

Logging status:

- CIS's benchmark recommends turning off syslog from receiving messages from the network at large. There are good reasons for this recommendation; it prevents an attacker from overwhelming syslogd or filling up the log filesystem. host-audit.sh does not check for this.

- In addition, the CIS benchmark represents turning on inetd logging. host-audit.sh concentrates on tcpd coverage instead, but this is incomplete. inetd logging should be turned on by running the "-t" option in /etc/init.d/inetd.

- cron logging should be enabled in /etc/default/cron by setting "CRONLOG=yes."

- CIS checks the permissions on important logs. In particular, the /var/adm/messages, utmpx and wtmpx files should not be writable by anyone other than root.

- loginlog should be created at /var/adm/loginlog, then make sure it is owned by root with no worse than 500 permissions.

- CIS recommends setting a limit on the number of failed login attempts in /etc/default/login.

- sar logging should be turned on in the sys crontab by uncommenting and/or editing the sa* entries.

- CIS also recommends examining the possibility of turning on kernel-level auditing. Since this can consume vast amounts of disk space, this should be done only when needed.

---

[16] Ingersoll, Wyllys. "Kerberos Network Security in the Solaris Operating Environment," Sun Microsystems, October 2001, http://www.sun.com/solutions/blueprints/1001/krb.pdf

[17] Reid, Jason and Watson, Keith. "Building and Deploying OpenSSH for the Solaris Operating Environment." Sun Microsystems, July 2001, http://www.sun.com/solutions/blueprints/0701/openSSH.pdf

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

- Process accounting can be turned on as well. This can sometimes be useful in tracking down odd behavior after the fact.

Serial Port login Disabled:

- Removing /usr/lib/saf/sac from /etc/inittab will keep the serial ports from presenting a login prompt. Note that this does not affect the console port, as that is run via a different mechanism.

Prevent Core Dumps:

- Setting sys:coredumpsize=0 in /etc/system will keep core dumps from tying up disk space and also prevent sensitive information from being exposed in a core dump. Unless the core dumps are actually going to be analyzed, they should be removed or prevented from occurring.

Set TCP Sequence Numbers:

- TCP_STRONG_ISS should be set to 2 in /etc/default/inetinit to foil people who are using sequence number guessing attacks.

File System Mounts:

- CIS recommends turning on ro and nosuid permissions wherever possible in the /etc/vfstab. UFS logging should also be enabled.

File/Directory Permissions:

- CIS checks this indirectly by seeing if the Sun fix-modes script has been run on the system, and also examines some files and directories directly. I have experimented with a few methods of checking file permissions directly, but I have not come up with one that I like yet.

  CIS makes pre-compiled binaries for fix-modes. They are located at ftp://ftp/CISecurity.org/pub/pkgs/Solaris/fix-modes.tar.Z

CDE and X Issues:

- CIS checks to see if Solaris 9 systems are having Xserver listen on port 9000. This can be turned off with the "-nolisten tcp" option in /etc/dt/config/Xservers. This prevents remote clients from displaying to the local system, but SSH tunneling can be used as an alternative.

- CIS recommends forcing the screensaver timeout to 10 seconds from the default of 30 seconds. To be honest, I am skeptical that it would make much difference.

Restrict at/cron Access:

- CIS recommends setting the /etc/cron.d/at.allow and cron.allow to only include people who should have access to cron and at facilities—preferably only root. the at.deny and cron.deny files should be removed at the same time.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Warning Banners:

- Warning banners should be set in /etc/motd, /etc/issue, /etc/default/telnetd, /etc/default/ftpd and the eeprom. These are equivalent to a no-trespassing sign.

Conclusion:

Designing and maintaining secure systems is a never-ending task. Systems that might have been considered secure a week ago become insecure with the release of the latest exploit. System administrators need to have flexible, easily customized tools to be able to keep up with ever-changing secure requirements.

check-advisories.sh, host-audit.sh and other security scripts only aid in setting a minimum level of system security. Application security, network security, physical security, and architectural security tend not to be easily examined by such scripts.

Perhaps the major contribution of these scripts is that they automate some of the drudge work of security engineering and allow more time for all the other challenges facing us.

© SANS Institute 2003, Author retains full rights

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

List of References:

- Center for Internet Security. "Solaris Benchmark v 1.1.0." Oct 2002. URL: https://www.cisecurity.org/tools2/solaris/SolarisBenchmark.pdf
- CERT. "Vulnerabilities, Incidents & Fixes." Carnegie Mellon Software Engineering Institute. Jan 2003. URL: http://www.cert.org/nav/index_red.html
- Cromar, Scott. "Solaris Security Advisories." Princeton University. May 2000. URL: http://www.princeton.edu/~cromar/SolarisSecurity/
- Cromar, Scott. "Sun Troubleshooting." Princeton University. Sep 2000. URL: http://www.princeton.edu/~unix/Solaris/troubleshoot/
- Radhakrishnan, Ramesh. "A Patch Management Strategy for the Solaris Operating Environment." Sun Microsystems. Jan 2003. URL: http://www.sun.com/solutions/blueprints/0103/817-1115.pdf.
- Security Focus. "Vulnerabilities by Vendor." Jan 2003. URL: http://online.securityfocus.com/bid
- Sun Microsystems. "Security Information: Security Bulletin Archive." Nov 2002. URL: http://sunsolve.sun.com/pub-cgi/secBulletin.pl
- Sun Microsystems. "SunSolve Patch Support Portal: Solaris OS Patch Reports." Jan 2003. URL: http://sunsolve.sun.com/pub-cgi/show.pl?target=patchpage
- Vöckler, Jens S. "Solaris 2.x-Tuning your TCP/IP Stack." URL: . http://www.sean.de/Solaris/soltune.html.
- Watson, Keith and Noordergraaf, Alex. "Solaris Operating Environment Network Settings for Security." Sun Microsystems. Dec 2000. URL: http://www.sun.com/solutions/blueprints/1200/network-updt1.pdf.
- Watson, Keith and Noordergraaf, Alex. "Solaris Operating Environment Security." Sun Microsystems. Dec 2002. URL: http://www.sun.com/solutions/blueprints/1202/816-5242.pdf.

Acknowledgements:
Thanks to Bill Bridges for his help in researching the security vulnerabilities for the original 1999 version of check.advisories.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Appendix A:
vuln-list-5.6

console:/kernel/misc/consconfig:105181:32:"SFBID 5161"
locale:/kernel/genunix:105181:25:"SFBID 1634, SSB 160, CA 97-18"
profil:/kernel/genunix:105181:18:"SFBID 570, SB 4261612"
kernel:/kernel/genunix:105181:33:"SB 4619870, SFBID 6309"
libc:/usr/lib/libc.a:105210:24:"SFBID 268, SSB 189"
libc:/usr/lib/libc.a:105210:36:"SB 4374039"
libc:/usr/lib/libc.a:105210:48:"SB 4311360, SB 4661997"
rpcbind:/usr/sbin/rpcbind:105216:04:"SFBID 67, SSB 167, SB 4124715, SB 4073327"
dtmail:/usr/dt/bin/dtmail:105338:27:"SFBID 3081, SFBID 175, SSB 181, SB 4166321, SB 4191180"
nfsserv:/kernel/misc/nfsserv:105379:06:"SB 4251026, SB 4072666"
sendmail:/usr/lib/sendmail:105395:05:"SSB 187, SB 4072035"
libnsl:/usr/lib/libnsl.a:105401:40:"SFBID 677, SFBID 148, SFBID 6484, SSB 172, SSB 170, SB 3405859, SFBID 5356, SB 4691127"
rpcbind:/usr/sbin/rpcbind:105401:41:"SB 4680691"
ypbind:/usr/lib/netsvc/yp/ypbind:105403:04:"SB 4362647"
libcurses:/usr/ccs/lib/libcurses.a:105405:03:"SB 4313067"
volrmmount:/usr/bin/volrmmount:105407:01:"SFBID 250, SSB 162, SB 4074650"
mutex_enter:/kernel/genunix:105529:07:"SFBID 655"
tcp:/kernel/drv/tcp:105529:15:"SB 4635484"
nisd_resolv:/usr/sbin/rpc.nisd_resolv:105552:03:"SB 4124715"
dtpad:/usr/dt/bin/dtpad:105558:02:"SB 1199005"
dtpad:/usr/dt/bin/dtpad:105558:03:"SB 4172128"
auth_des:/kernel/misc/rpcsec:105564:05:"SFBID 6484, SB 4240833"
rpc.cmsd:/usr/openwin/bin/rpc.cmsd:105566:15:"SFBID 575, SFBID 524, SFBID 166, SSB 188, SSB 183, SB 4117202, CA 99-08, SB 4203585"
mountd:/usr/lib/nfs/mountd:105615:08:"SFBID 95, SSB 168, SB 4124715"
Xsun:/usr/openwin/bin/Xsun:105633:62:"SFBID 4408, SFBID 2561, SFBID 1140, SB 4661987, SB 4703884, SB 4710402"
xlock:/usr/openwin/bin/xlock:105633:60:"SFBID 3160"
login:/bin/login:105665:04:"SFBID 3681, SFBID 437, SSB 213, CA 2001-34, SBB 4516885"
rdist:/bin/rdist:105667:03:"SFBID 129, SSB 179, SB 4284268, SB 4119069, SB 4072602"
libDtSvc.so.1:/usr/lib/libDtSvc.so.1:105669:11:"SFBID 3517, SFBID 3147, SSB 214, SSB 164, SB 4527363, SB 4191060, SB 4274081, CA 2001-31"
dtlogin:/usr/dt/bin/dtlogin:105703:26:"SB 4274081, SB 4179987, SB 4274081"
ufsrestore:/usr/lib/fs/ufs/ufsrestore:105722:06:"SFBID 1348, SFBID 680, SSB 210, SSB 189, SB 4339366, SB 4132365, VU 36866"
bind:/usr/sbin/in.named:105755:10:"SFBID 788, SSB 204, SSB 180, SB 4409676, CA 2001-02"
resolver:/usr/lib/libresolv.so.2:105755:12:"CA 2002-19, SB 4708913, SFBID 5100"
fifofs:/kernel/fs/fifofs:105780:03:"SB 4149694, SB 4090929"
ndd:/kernel/drv/ip:105786:01:"SFBID 433, SSB 165"
ip:/kernel/drv/ip:105686:11:"SB 4184794, SB 4242224"
admintool:/bin/admintool:105800:07:"SFBID 4624, SB 4354306, SB 4304397"

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

CDE:/usr/dt/bin/dtsession:105802:16:"SFBID 3382, SFBID 122, SSB 212, SSB 192, SB 4203589, SB 4164808, CA 2001-27, CA 99-11"
Tooltalk:/usr/openwin/bin/rpc.ttdbserverd:105802:19:"SB 4713445, SB 4707187, SFBID 5444, SFBID 5083, SFBID 5082"
CDE:/usr/dt/bin/dtsession:105837:02:"SSB 185, CA 98-02, SB 4117696"
dtappgather:/usr/dt/bin/dtappgather:105837:03:"SFBID 131, SB 4097549"
se:/kernel/drv/se:105924:17:"SB 46575034, SB 4685756"
vi:/usr/bin/vi:105990:05:"SB 4364594, SB 4161925"
dtsession:/usr/dt/bin/dtsession:106027:10:"SFBID 2063, SB 4448598"
ximp40:/usr/dt/bin/dtaction:106040:17:"SFBID 2322, SB 4409148"
telnetd:/usr/sbin/in.telnetd:106049:04:"SFBID 3064, SFBID 5848, SFBID 5531, SB 4082063, SB 4483514, SB 4523990, SB 4527873"
CDE:/usr/dt/bin/dtsession:106112:03:"SSB 185, CA 98-02, SB 4081672"
CDE:/usr/dt/bin/sdtvolcheck:106112:06:"SB 4255264"
catman:/bin/catman:106123:05:"SFBID 2149, SB 4392144"
man:/bin/man:106123:05:"SFBID 165, SSB 184, SB 4392144"
patchadd:/usr/sbin/patchadd:106125:13:"SFBID 2127, SB 4399797, SB 4500354"
ff.core:/usr/openwin/bin/ff.core:106222:01:"SB 4114295"
lp:/bin/lp:106235:06:"SFBID 1200, SFBID 1138, SFBID 1143, SFBID 251, SSB 206, SSB 195, SB 4334568"
lpd:/usr/lib/print/in.lpd:106235:10:"SFBID 3274, SFBID 2894, SSB 206, CA 2001-15, SB 4501950"
dtprintinfo:/usr/dt/bin/dtprintinfo:106242:03:"SFBID 4630, SB 4345282"
passwd:/bin/passwd:106257:06:"SFBID 174, SSB 182, SB 4284795, SB 4223215, SB 1236638"
pkgadd:/usr/sbin/pkgadd:106292:13:"SFBID 5208, SB 4136905"
ftpd:/usr/sbin/in.ftpd:106301:03:"SFBID 2564, SFBID 2550, SSB 205, SSB 171, SB 4436988, SB 4446600, CA 2001-07"
yppasswdd:/usr/lib/netsvc/yp/rpc.yppasswdd:106303:03:"SFBID 2763, SSB 209, VU 327281, SB 4456994, SB 4392250"
Xview:/usr/openwin/lib/libxview.a:106331:05:"SB 4458476"
dtprintinfo:/usr/dt/bin/dtprintinfo:106437:03:"SFBID 4630, SFBID 249, SB 4139394"
sh:/sbin/sh:106361:14:"SB 4392404, SB 4494351, SB 4384080, SB 4404641"
xdm:/usr/openwin/bin/xdm:106415:04:"SB 4388773"
dtprintinfo:/usr/dt/bin/dtprintinfo:106437:03:"SB 4139394, SB 4191060, SB 4007233, SB 4191065"
ping:/usr/sbin/ping:106448:01:"SSB 174, SB 4074562"
uucp:/bin/uucp:106468:04:"SFBID 2253, SB 4406722, SB 4416701"
ftp:/usr/bin/ftp:106522:03:"SFBID 2601, SFBID 178, SSB 176, SB 4197316"
libauth:/usr/lib/libauth.a:106569:01:"SFBID 442, SB 4157655"
rpc.statd:/usr/lib/nfs/statd:106592:03:"SFBID 450, SSB 186, SB 4124715, SB 4159085"
libsec:/usr/lib/libsec.a:106625:13:"SB 4152786, SB 4300951"
libce:/usr/openwin/lib/libce.a:106648:01:"SSB 175, CA 98-10"
rpcmod:/kernel/sys/rpcmod:106639:07:"SB 4691127"
libce:/usr/openwin/lib/libce.a:106648:01:"SB 4153830"
libdeskset:/usr/openwin/lib/libdeskset.so.0:106649:01:"SSB 175, CA 98-10, SB 4153829"

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

mailtool:/usr/openwin/bin/mailtool:106650:04:"SFBID 2787, SSB 175, SB 4296268, SB 4153829, CA 98-10"
mibiisa:/usr/lib/dmi/dmispd:106787:18:"SFBID 4933, SFBID 2417, SSB 215"
snmpdx:/usr/lib/snmp/snmpdx:106787:18:"SFBID 4932, SFBID 4088, SFBID 4089, SSB 215, SSB 207, CA 2002-03, CA 2001-05"
mv:/usr/bin/mv:106834:01:"SB 4140373"
ntpd:/usr/local/bin/ntpd:107298:03:"SFBID 2540, SSB 211, VU 970472, SB 4434235"
telmod:/usr/kernel/telmod:107326:01:"SB 4448655"
kcms:/usr/openwin/bin/kcms_server:107336:01:"SFBID 2605, SFBID 452, SB 4199722"
tftpd:/usr/sbin/in.tftpd:107565:02:"SB 4254347"
vold:/usr/sbin/vold:107618:04:"SFBID 5207, SFBID 327, SB 4637250, SB 4194660"
pgxconfig:/usr/sbin/pgxconfig:107715:18:"SFBID 5390"
ld.so:/usr/lib/ld.so:107733:01:"SFBID 659, SB 4150646, SB 4448531"
ld.so:/usr/lib/ld.so:107733:10:"SB 4433643, SB 4488954"
pax:/bin/pax:107758:02:"SB 4119120, SB 4061087"
inetd:/usr/sbin/inetd:107774:01:"SB 4154509"
pgxconfig:/usr/sbin/pgxconfig:107851:19:"SFBID 5390"
rcp:/bin/rcp:107991:01:"SFBID 268, SSB 189, SB 4240566"
xfs:/usr/openwin/bin/xfs:108129:05:"SFBID 6241, SB 4764193, CA-2002-34"
dtspcd:/usr/dt/bin/dtspcd:108199:01:"SFBID 2322, SSB 636, SSB 192, SB 4257351"
dtaction:/usr/dt/bin/dtaction:108201:01:"SFBID 635, SSB 192, CA 99-11, SB 4257350"
keyserv:/usr/sbin/keyserv:108307:02:"SB 4124715"
jserver:/usr/lib/locale/ja/wnn/jserver_m:108333:02:"SB 4352777"
rpc.nispasswdd:/usr/sbin/rpc.nispasswdd:108346:03:"SB 4124715"
ldterm:/kernel/strmod/ldterm:108468:01:"SB 4102102"
snoop:/usr/sbin/snoop:108492:01:"SFBID 864, SSB 190, SB 4282985"
sadmind:/usr/sbin/sadmind:108660:01:"SFBID 866, SFBID 2354, SSB 191, CA 99-16, SB 4298053"
tip:/usr/bin/tip:108804:02:"SFBID 2475, SB 4330475, SB 4063098, SB 4430971"
ypxfrd:/usr/lib/netsvc/yp/ypxfrd:108890:02:"SB 4737417, SB 4124715"
rpc.ypupdated:/usr/lib/netsvc/yp/rpc.ypupdated:108893:01:"SB 4124715"
rpc.bootparamd:/usr/sbin/rpc.bootparamd:108895:01:"SB 4124715"
mkdevmaps:/sr/sbin/mkdevmaps:109100:02:"SB 1129659"
mail:/bin/mail:109266:05:"SFBID 4107, SB 4705717, SB 4624990, SB 4465086"
nscd:/usr/sbin/nscd:109339:01:"SB 4114757"
chkperm:/usr/vmsys/bin/chkperm:109388:01:"SFBID 918, SB 4252402, SB 4296167, SB 4303199"
arp:/usr/sbin/arp:109719:01:"SFBID 2193, SFBID 837, SSB 200, SB 4296166"
useradd:/usr/sbin/useradd:110883:01:"SB 4222400"
ttymon:/usr/lib/saf/ttymon:110990:02:"SB 4657339"
sdiff:/usr/sbin/sdiff:111039:02:"SB 4064007"
finger:/usr/sbin/in.fingerd:111236:01:"SFBID 3457, SB 4298915"
finger:/bin/finger:111240:01:"SFBID 3457, SB 4298986"
dmesg:/bin/dmesg:111560:01:"SB 4110589"
libmle:/usr/4lib/libmle.so.1.4:111645:01:"SB 4468138"
whodo:/usr/sbin/whodo:111859:01:SFBID 2935, SB 4477380"

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

mailx:/usr/bin/mailx:112073:02:"SFBID 2610, SFBID 2169, SFBID 1910, SFBID 393, SB 4452732"
pt_chmod:/usr/lib/pt_chmod:112456:01:"SB 4448407"
vipw:/usr/ucb/vipw:112765:01:"SB 4198184"
pcmciad:/usr/lib/pcmciad:112792:01:"SFBID 5268"
talkd:/usr/sbin/in.talkd:112814:01:"SB 4651310"
rwall:/usr/lib/netsvc/rwall/rpc.rwalld:112893:01:"SFBID 4639, SB 4664537"
utmp_update:/usr/lib/utmp_update:113754:01:"SB 4705891"

© SANS Institute 2003, Author retains full rights

Appendix B:
vuln-list-5.8

locale:/kernel/genunix:108528:15:"SFBID 1634"
poll:/kernel/genunix:108528:15:"SFBID 5171"
kernel:/kernel/genunix:108528:16:"SFBID 6080"
kernel:/kernel/genunix:108528:17:"SFBID 6309"
ip:/kernel/drv/ip:108528:18:"SB 4691577, SFBID 6147"
Xsun:/usr/openwin/bin/Xsun:108652:53:"SFBID 4408, SFBID 2561, SFBID 1140, SB 4661987"
lbxproxy:/usr/openwin/bin/lbxproxy:108652:51:"SFBID 4633, SB 4649617"
xlock:/usr/openwin/bin/xlock:108652:59:"SFBID 3160, SB 4703884"
ximp40:/usr/dt/bin/dtaction:108773:14:"SFBID 2322, SB 4409148, SB 4524098"
xsunim_adaptor:/usr/lib/im/leif/xsunimadapter.so:108773:17:"SB 4777933"
libc:/usr/lib/libc.a:108827:15:"SB 4374039"
libnsl:/usr/lib/libnsl.a:108827:31:"SB 3405859, SFBID 5356, SB 4691127"
rpc:/usr/lib/libc.a:108827:36:"SB 4680691"
rpc.cmsd:/usr/dt/bin/rpc.cmsd:108835:03:"SB 4641721"
mibiisa:/usr/lib/dmi/dmispd:108869:16:"SFBID 4933, SFBID 2417, SB 46369285 SB 4639509, SSB 215"
snmpdx:/usr/lib/snmp/snmpdx:108869:16:"SFBID 4932, SFBID 4088, SFBID 4089, SB 4640230, SB 4639581, SSB 215, SSB 207, CA 2002-03, CA 2001-05"
rpcmod:/kernel/sys/rpcmod:108901:06:"SB 4691127"
smartcard:/usr/dt/bin/sdtsmartcardadmin:108909:06:"SB 4343393"
dtsession:/usr/dt/bin/dtsession:108919:15:"SFBID 2063, SB 4448598"
dtprintinfo:/usr/dt/bin/dtprintinfo:108949:04:"SFBID 4630"
dtspcd:/usr/dt/bin/dtspcd:108949:07:"SFBID 3517, SSB 214, CA 2001-31"
libDtSvc:/usr/dt/lib/libDtSvc.so.1:108949:07:"SFBID 3147, SB 4191060, SB 4191060, SB 4527363"
vold:/usr/sbin/vold:108968:07:"SFBID 5207, SB 4637250"
sd:/kernel/drv/sd:108974:22:"SB 4673801"
rmformat:/usr/bin/rmformat:108975:02:"SB 4322206"
rshd:/usr/sbin/in.rshd:108985:02:"SB 4335632"
patchadd:/usr/sbin/patchadd:108987:08:"SFBID 2127, SB 4500354"
libsldap:/usr/include/ldap.h:108993:05:"SFBID 2931"
getgrent:/usr/sbin/ldapclient:108993:11:"SB 4614945"
su:/usr/bin/su:109005:04:"SB 4411652"
dhcpd:/usr/sbin/dhcpconfig:109077:02:"SB 4123989"
dhcpd:/usr/sbin/dhcpconfig:109077:10:"SB 4607109"
ufsrestore:/usr/lib/fs/ufs/ufsrestore:109091:05:"SFBID 1348, SSB 210, SB 4339366, SB 4132365, VU 36866, SB 4498121"
wbem:/etc/security/exec_attr:109134:27:"SFBID 6061, SB 4381755, SB 4500475, SB 4417342"
mkdevalloc:/usr/sbin/mkdevalloc:109149:02:"SB 1229659"
pgxconfig:/usr/sbin/pgxconfig:109154:14:"SFBID 5390, SB 4728662"
gld:/kernel/misk/gld:109202:03:"SB 4467926"
apache:/usr/apache/bin/httpd:109234:09:"SB 4705227"
ipcs:/usr/xpg4/bin/ipcs:109238:01:"SB 4310353"

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

lp:/bin/lp:109320:01:"SFBID 1200, SFBID 1143, SFBID 1138, SSB 206, SSB 195"
lpd:/usr/lib/print/in.lpd:109320:05:"SFBID 3274, SFBID 2894, SSB 206, SB 4501950, SB 4504977, SSB 206, CA 2001-15"
sh:/sbin/sh:109324:03:"SB 4392404"
bind:/usr/sbin/in.named:109326:03:"SB 4409676"
bind:/usr/sbin/in.named:109326:04:"SFBID 788, SSB 204, CA 2001-02"
resolver:/usr/lib/libresolv.so.2:109326:09:"CA 2002-19, SB 4708913, SB 4646349, SFBID 5100"
ypxfrd:/usr/lib/netsvc/yp/ypxfrd:109328:03:"SB 4737417"
dtsession:/usr/dt/bin/dtsession:109354:13:"SFBID 2063, SB 4430559, SB 4489859, SB 4448598, SB 4489859"
ntpd:/usr/local/bin/ntpd:109667:04:"SFBID 2540, SB 4434235, SSB 211, VU 970472"
smartcard:/etc/smartcard/opencard.properties:109695:03:"SB 4343216"
lockd:/usr/lib/nfs/lockd:109783:02:"SB 4492876, SB 4325431, SFBID 5986"
su:/platform/sun4u/kernel/drv/su:109793:13:"SB 4587859, SB 4666211, SB 4464201"
pam_krb5:/usr/lib/security/pam_kerb5.so.1:109805:05:"SB 4499330, SB 4373142, SB 4351689"
se:/kernel/drv/se:109815:15:"SB 4587859"
xfs:/usr/openwin/lib/fs.auto:109862:03:"SFBID 6241"
smartcard:/usr/sbin/ocfserv:109887:14:"SB 4524620, SB 4629775, SB 4628969"
picld:/usr/lib/llib-lpicl:109888:12:"SB 4417600"
ecpp:/kernel/drv/ecpp:109892:02:"SB 4364900"
stc:/usr/include/sys/stcio.h:109893:03:"SB 4321509"
stc:/usr/include/sys/stcio.h:109893:04:"SB 4587859"
usb:/kernel/drv/usbprn:109896:11:"SB 4587859"
arp:/kernel/drv/arp:109898:05:"SB 4365204, SB 4363786, SB 4427290"
jserver:/usr/lib/locale/ja/wnn/jserver_m:"109951:01:"SB 4352777"
sdtpdasync:/usr/dt/appconfig/sdtpdasync/classes/SyncMgr.jar:110068:02:"SB 4367008"
devinfo:/usr/sbin/devinfo:110075:01:"SB 4341354"
ttdbserverd:/usr/openwin/bin/rpc.ttdbserverd:110286:10:"SFBID 3382, SSB 212, CA 2001-27, SB 473445, SB 4707187, SFBID 5444, SFBID 5598, SFBID 5083, SFBID 5082"
ypbind:/usr/lib/netsvc/yp/ypbind:110322:01:"SB 4362647"
getexecuser:/etc/security/exec_attr:110386:02:"SB 4458070"
ufsdump:/usr/lib/fs/ufs/ufsdump:110387:02:"SB 4339366, SB 4132365"
atok:/usr/sbin/atok12mngtool:110416:03:"SB 4443974, SB 4361738, SB 4372858"
admintool:/bin/admintool:110453:03:"SFBID 4624, SB 4354306, SB 4304397"
libcurses:/usr/ccs/lib/libcurses.a:110458:01:"SB 4313067"
picld:/usr/lib/llib-lpicl:110460:12:"SB 4417600"
sendmail:/usr/lib/sendmail:110615:01:"SB 4072035"
telnetd:/usr/sbin/in.telnetd:110668:03:"SFBID 3064, SFBID 5848, SFBID 5531, SB 4366956, SB 4483514, SB 4523990, SB 4527873, SB 4516876"
rcp:/usr/sbin/static/rcp:110670:01:"SB 4366956"
dman:/platform/SUNW,Sun-Fire-15000/kernel/drv/sparcv9/dman:110820:10:"SB 4587859"
cachefsd:/usr/lib/fs/cachefs/cachefsd:110896:02:"SFBID 4674, SB4338920, SB 4467621"
csh:/usr/bin/csh:110898:01:"SB 4384080"
csh:/usr/bin/csh:110898:04:"SB 4494351, SB 4404641"
vi:/usr/bin/vi:110903:02:"SB 4364594"

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

pkgadd:/usr/sbin/pkgadd:110934:08:"SFBID 5208, SB 4136905"
tcsh:/usr/bin/tcsh:110943:01:"SB 4384076"
llc2:/usr/kernel/drv/llc2:110953:04:"SB 4587859"
timod:/kernel/drv/timod:110955:04:"SB 4587859"
mailx:/usr/bin/mailx:110957:02:"SFBID 2610, SFBID 2169, SFBID 1910, SB 4452732"
bsmunconv:/etc/security/bsmunconv:111069:01:"SB 4383308"
cu:/usr/bin/cu:111071:01:"SB 4406722"
login:/bin/login:111085:02:"SFBID 3681, SSB 213, CA 2001-34, SB 4516885"
finger:/usr/sbin/in.fingerd:111232:01:"SFBID 3457, SB 4298915"
finger:/usr/bin/finger:111234:01:"SFBID 3457, SB 4298986"
ppp:/usr/sbin/aspppd:111299:04:"SB 4683015"
klmmod:/kernel/misc/klmmod:111321:03:"SB 4492876, SFBID 5986"
ttymon:/usr/lib/saf/ttymon:111325:02:"SB 4657339"
dcs:/usr/lib/dcs:111332:04:"SB 4480306"
kcms:/usr/openwin/bin/kcms_server:111400:01:"SFBID 2605, SFBID 2558, SB 4415570"
tip:/usr/bin/tip:111504:01:"SFBID 2475, SB 4330475, SB 4430971"
catman:/bin/catman:111548:01:"SFBID 2149, SB 4392144"
uucp:/bin/uucp:111570:01:"SFBID 2253, SB 4416701"
specfs:/kernel/drv/specfs:111588:04:"SB 4587859"
yppasswdd:/usr/lib/netsvc/yp/rpc.yppasswdd:111596:02:"SFBID 2763, SSB 209, VU 327281, SB 4456994"
ftpd:/usr/sbin/in.ftpd:111606:02:"SFBID 2564, SFBID 2550, SFBID 2496, SSB 205, SB 4436988, SB 4451524, CA 2001-07"
inetd:/usr/sbin/inetd:111624:04:"SB 4383820"
Xview:/usr/openwin/lib/libxview.a:111626:01:"SFBID 5479, SB 4458476"
libmle:/usr/4lib/locale/ja/libmle.so.1.4:111647:01:"SB 4468138"
passwd:/usr/bin/passwd:111659:02:"SB 4112707"
ftpd:/usr/bin/passwd:111659:07:"SB 4393399, SB 4450103"
PAM:/usr/lib/libpam.so.1:111659:07:"SFBID 5269, SB 4393399"
whodo:/usr/sbin/whodo:111826:01:"SFBID 2935, SB 4477380"
telmod:/usr/kernel/strmod/telmod:111881:03:"SB 4587859"
mail:/bin/mail:111874:05:"SFBID 4107, SB 4624990"
ckitem:/usr/bin/ckitem:112039:01:"SB 4466215"
pam_ldap:/usr/lib/security/pam_ldap.so.1:112218:01:"SB 4384816"
pt_chmod:/usr/lib/pt_chmod:112459:01:"SFBID 3522"
kerberos:/usr/lib/gss/gl/mech_krb5.so.1:112237:07:"SB 4691352"
kerberos:/usr/lib/gss/do/mech_krb5.so.1:112390:09:"SB 4691352"
random:/etc/devlink.tab:112438:01:"SB 4337350"
pt_chmod:/usr/lib/pt_chmod:112459:01:"SB 4394991"
autofs:/kernel/fs/autofs:112605:02:"SB 4631449"
autofs:/kernel/fs/autofs:112605:04:"SB 4525971"
libz:/usr/lib/libz.so.1:112611:01:"SB 4644859"
gzip:/usr/bin/gzip:112668:01:"SB 4644742"
rwall:/usr/lib/netsvc/rwall/rpc.rwalld:112846:01:"SFBID 4639"
libz:/usr/lib/libz.so.1:112611:01:"SB 4644859"
gzip:/usr/bin/gzip:112668:01:"SB 4644742"
pcmciad:/usr/lib/pcmciad:112792:01:"SB 4280870"

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

talkd:/usr/sbin/in.talkd:112796:01:"SB 4651310"
rpc.rwalld:/usr/lib/netsvc/rwall/rpc.rwalld:112846:01:"SB 4664537"
utmp_update:/usr/lib/utmp_update:113650:01:"SB 4705891"
llc1:/kernel/drv/llc1:113685:01:"SB 4587859"
kbtrans:/kernel/misc/kbtrans:113687:01:"SB 4587859"
mailtool:/usr/openwin/bin/mailtool:113792:01:"SB 4755258"

© SANS Institute 2003, Author retains full rights

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Appendix C:
vuln-list-5.9

kernel:/kernel/genunix:112233:02:"SB 4691127, SFBID 5356, SFBID 6309"
rpc.cmsd:/usr/dt/bin/rpc.cmsd:112617:01:"SB 4641721"
ttdbserverd:/usr/openwin/bin/rpc.ttdbserverd:112808:03:"SB 4713445, SB 4707187, SFBID 5082, SFBID 5444, SFBID 5083, SFBID 5082"
libc:/usr/lib/libc.a:112874:01:"SB 4661997"
rpc.rwalld:/usr/lib/netsvc/rwall/rpc.rwalld:112875:01:"SB 4664537"
IP:/kernel/drv/ip:112902:07:"SB 4691577, SB 4724336, SFBID 6147"
krb5:/kernel/misc/kgss/gl_kmech_krb5:112908:04:"SB 4691352, SB 4690212"
smartcard:/etc/smartcard/opencard.properties:112926:03:"SB 4629775"
wbem:/usr/sadm/lib/wbem.jar:112945:03:"SB 4699585"
resolver:/usr/lib/libresolv.so.2:112970:02:"CA 2002-19, SB 4708913, SFBID 6147"
doorfs:/kernel/sys/doorfs:113030:01:"SB 4659950"
apache:/usr/apache/bin/httpd:113146:01:"SB 4705227"
sshd:/usr/lib/ssh/sshd:113273:01:"SB 4708590"
lockd:/usr/lib/nfs/lockd:113278:01:"SB 4492876, SFBID 5986"
klmmod:/kernel/misc/klmmod:113279:01:"SB 4492875, SFBID 5986"
libnsl:/usr/lib/libnsl.a:113319:01:"SB 4691127, SFBID 5356"
sendmail:/usr/lib/sendmail:113575:01:"SB 4704675, SB 4704672"
ypxfrd:/usr/lib/netsvc/yp/ypxfrd:113579:01:"SB 4737417"
utmp:/usr/lib/utmp_update:113718:01:"SB 4705891"
xfs:/usr/openwin/bin/xfs:113923:02:"CA-2002-34, SFBID 6241"

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Appendix D:
netconfig

```
#!/bin/sh
# Script to set network security parameters as per Sun recommendations
# SCC 30 July 2001

OS=`uname -sr | awk '{print $2}'`

case "${OS}" in
  5.6|5.7)
      ndd -set /dev/ip ip_forwarding 0
      ndd -set /dev/ip ip_strict_dst_multihoming 1
      ndd -set /dev/ip ip_send_redirects 0
      ndd -set /dev/ip ip_ignore_redirect 1
      ndd -set /dev/ip ip_forward_src_routed 0
      ndd -set /dev/ip ip_forward_directed_broadcasts 0
      ndd -set /dev/tcp tcp_conn_req_max_q0 4096
      ndd -set /dev/tcp tcp_ip_abort_cinterval 60000
      ndd -set /dev/ip ip_respond_to_timestamp 0
      ndd -set /dev/ip ip_respond_to_timestamp_broadcast 0
      ndd -set /dev/ip ip_respond_to_address_mask_broadcast 0
      ndd -set /dev/arp arp_cleanup_interval 60000
      ndd -set /dev/ip ip_ire_flush_interval 60000
  ;;
    5.8|5.9)
      ndd -set /dev/ip ip_forwarding 0
      ndd -set /dev/ip ip6_forwarding 0
      ndd -set /dev/ip ip_strict_dst_multihoming 1
      ndd -set /dev/ip ip6_strict_dst_multihoming 1
      ndd -set /dev/ip ip_send_redirects 0
      ndd -set /dev/ip ip6_send_redirects 0
      ndd -set /dev/ip ip_ignore_redirect 1
      ndd -set /dev/ip ip6_ignore_redirect 1
      ndd -set /dev/ip ip_forward_src_routed 0
      ndd -set /dev/ip ip6_forward_src_routed 0
      ndd -set /dev/ip ip_forward_directed_broadcasts 0
      ndd -set /dev/tcp tcp_conn_req_max_q0 4096
      ndd -set /dev/tcp tcp_ip_abort_cinterval 60000
      ndd -set /dev/ip ip_respond_to_timestamp 0
      ndd -set /dev/ip ip_respond_to_timestamp_broadcast 0
      ndd -set /dev/ip ip_respond_to_address_mask_broadcast 0
      ndd -set /dev/arp arp_cleanup_interval 60000
      ndd -set /dev/ip ip_ire_arp_interval 60000
  ;;
esac
```

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Last Updated: May 25th, 2013

Upcoming SANS Training

Click Here for a full list of all Upcoming SANS Events by Location

SANS Malaysia @ MCMC 2013	Cyberjaya, MY	Jun 03, 2013 - Jun 08, 2013	Live Event
SANS Pen Test Berlin 2013	Berlin, DE	Jun 03, 2013 - Jun 08, 2013	Live Event
Industial Control Systems Security Training - Houston	Houston, TXUS	Jun 10, 2013 - Jun 15, 2013	Live Event
Security Impact of IPv6 Summit 2013	Washington, DCUS	Jun 14, 2013 - Jun 16, 2013	Live Event
SANSFIRE 2013	Washington, DCUS	Jun 14, 2013 - Jun 22, 2013	Live Event
SANS Canberra 2013	Canberra, AU	Jul 01, 2013 - Jul 13, 2013	Live Event
Digital Forensics & Incident Response Summit 2013	Austin, TXUS	Jul 09, 2013 - Jul 16, 2013	Live Event
SANS London Summer 2013	London, GB	Jul 09, 2013 - Jul 16, 2013	Live Event
SANS Rocky Mountain 2013	Denver, COUS	Jul 14, 2013 - Jul 20, 2013	Live Event
SANS Mumbai 2013	Mumbai, IN	Jul 22, 2013 - Jul 27, 2013	Live Event
SEC528 SANS Training Program for the CompTIA® New Advanced Security Practitioner (CASP) Certification	Washington, DCUS	Jul 22, 2013 - Jul 26, 2013	Live Event
SANS San Francisco 2013	San Francisco, CAUS	Jul 29, 2013 - Aug 03, 2013	Live Event
SANS SEC 560: Network Penetration Testing @ Bangalore 2013	Bangalore, IN	Aug 05, 2013 - Aug 10, 2013	Live Event
SANS Boston 2013	Boston, MAUS	Aug 05, 2013 - Aug 10, 2013	Live Event
Critical Security Controls Summit	Washington, DCUS	Aug 12, 2013 - Aug 18, 2013	Live Event
Industrial Control Systems Security Training - DC	Washington, DCUS	Aug 12, 2013 - Aug 16, 2013	Live Event
SANS Thailand 2013	Bangkok, TH	Aug 19, 2013 - Aug 31, 2013	Live Event
SANS Virginia Beach 2013	Virginia Beach, VAUS	Aug 19, 2013 - Aug 30, 2013	Live Event
Mobile Device Security Summit 2013	OnlineCAUS	May 30, 2013 - Jun 06, 2013	Live Event
SANS OnDemand	Books & MP3s OnlyUS	Anytime	Self Paced