SOLARIS 8 BUILD DOCUMENT

TABLE OF CONTENTS

SOLARIS 8 BUILD DOCUMEN T

AUTHORED BY:

Gideon Rasmussen, CISSP
Information Security Manager
Infostruct L.L.C.
Norwalk, CT
gideon@infostruct.net

SYSTEM CONFIGURATIONS

Purpose

This document details the configuration, hardening, monitoring and vulnerability assessment of the Solaris operating system. It can also be used as a configuration standard, providing a baseline to audit against. It is important to understand the configurations at a granular level to troubleshoot outages. Installs and hardening can be automated with Jumpstart and the Solaris Security Toolkit (respectively).

Installed Solaris 8

Installed Solaris 8 using the following file systems:

File System	Size	Partition
/ (root)	4 GB	c0t0d0s0
swap	See below	c0t0d0s1
/usr	4 GB	c0t0d0s3
/var	4 MB	c0t0d0s4
/opt	7 MB	c0t0d0s5
/export/home	5 GB	c0t0d0s6
/app	12 MB	c0t0d0s7

Swap should be equal to twice the size of the memory installed on the server. To determine the amount of system memory, use "*/usr/platform/sun4u/sbin/prtdiag –v*".

Volume Manager configurations are outside of the scope of this document.

Installed a Patch Cluster

Installed the latest recommended and security patch cluster from http://sunsolve.sun.com. Searched for hardware specific patches as well.

```
# cd /tmp
# unzip 8_Recommended.zip
# cd 8_Recommended
# ./install_cluster
# /usr/sbin/shutdown –i6 –g0 –y
```

Enabled DNS

```
# vi /etc/nsswitch.conf

hosts:      files dns

# vi /etc/resolv.conf
domain domain.com
nameserver 192.168.1.105
nameserver 192.168.1.106
search domain.com
```

Configured the Default Gateway

Configured on-line:

```
# route add net default 192.168.1.1 1
```
("1" at the end signifies how many hops. It should be set to 1 because the first thing the server hits is the NIC card)

Configured for reboot:

```
# vi /etc/defaultrouter
192.168.1.1
```

Added FQDN to /etc/hosts

```
# vi /etc/hosts

192.168.1.101     sunsrv01.domain.com sunsrv01 loghost
```

Added fully qualified domain name to /etc/hosts to prevent sendmail errors (My unqualified host name (*hostname*) unknown; sleeping for retry)

Added Additional IP Addresses

```
# vi /etc/hosts
192.168.1.15     projqa
192.168.1.16     projdev
```

```
# vi hostname.eri0:1
projqa
# vi hostname.eri0:2
projdev
# ifconfig eri0:1 plumb
# ifconfig eri0:1 inet 192.168.1.15 broadcast 192.168.1.255 netmask 255.255.255.0 –
trailers
# ifconfig eri0:1 up
# ifconfig eri0:2 plumb
# ifconfig eri0:2 inet 192.168.1.16 broadcast 192.168.1.255 netmask 255.255.255.0 –
trailers
# ifconfig eri0:2 up
# ifconfig -a
lo0: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
eri0: flags=1000863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 192.168.1.14 netmask ffffff00 broadcast 192.168.1.255
        ether 0:3:ba:b:3:f5
eri0:1: flags=1000863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 192.168.1.15 netmask ffffff00 broadcast 192.168.1.255
eri0:2: flags=1000863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 192.168.1.16 netmask ffffff00 broadcast 192.168.1.255
```

Forced NIC to 100 Mbps Full Duplex

To prevent issues with auto negotiation, forced both the network switch and the system's NIC cards to 100 Mbps, full duplex.

Determined which interfaces are available:

```
# ifconfig -a
```

Configured on-line (use only the interfaces found on the system):

hme:

```
# ndd -set /dev/hme instance 0
# ndd -set /dev/hme adv_100fdx_cap 1
# ndd -set /dev/hme adv_autoneg_cap 0
```

qfe:

```
# ndd -set /dev/qfe instance 0
# ndd -set /dev/qfe adv_100fdx_cap 1
# ndd -set /dev/qfe adv_autoneg_cap 0
```

eri:

```
# ndd -set /dev/eri instance 0
# ndd -set /dev/eri adv_100fdx_cap 1
# ndd -set /dev/eri adv_autoneg_cap 0
```

ce:

```
# ndd -set /dev/ce instance 0
# ndd -set /dev/ce link_master 0
# ndd -set /dev/ce adv_1000fdx_cap 0
# ndd -set /dev/ce adv_1000hdx_cap 0
# ndd -set /dev/ce adv_100fdx_cap 1
```

```
# ndd -set /dev/ce adv_100hdx_cap 0
# ndd -set /dev/ce adv_10fdx_cap 0
# ndd -set /dev/ce adv_10hdx_cap 0
# ndd -set /dev/ce adv_autoneg_cap 0
```

Configured for reboot (use only the interfaces found on the system):

hme, qfe & eri:

```
# vi /etc/system   (ensure there are no blank lines)
* Force hme into 100 Mbps full duplex mode
set hme:hme_adv_100fdx_cap=1
* Don't negotiate operation mode with the network hub
set hme:hme_adv_autoneg_cap=0
* Force qfe into 100 Mbps full duplex mode
set qfe:qfe_adv_100fdx_cap=1
* Don't negotiate operation mode with the network hub
set qfe:qfe_adv_autoneg_cap=0
* Force eri into 100 Mbps full duplex mode
set eri:adv_100fdx_cap=1
* Don't negotiate operation mode with the network hub
set eri:adv_autoneg_cap=0
```

ce:

```
# vi /etc/rc2.d/S99net-tune
#!/sbin/sh
# Set NIC to 100 Mbps full duplex
ndd -set /dev/ce instance 0
ndd -set /dev/ce link_master 0
ndd -set /dev/ce adv_1000fdx_cap 0
ndd -set /dev/ce adv_1000hdx_cap 0
ndd -set /dev/ce adv_100fdx_cap 1
ndd -set /dev/ce adv_100hdx_cap 0
ndd -set /dev/ce adv_10fdx_cap 0
ndd -set /dev/ce adv_10hdx_cap 0
ndd -set /dev/ce adv_autoneg_cap 0
exit 0

# chmod 700 /etc/rc2.d/S99net-tune
```

Confirmed settings:

hme, qfe and eri:

```
# ifconfig -a
```

ce:

```
# netstat -k ce0 | grep link_speed
link_speed 100 link_duplex 2 link_asmpause 0 link_pause 0
```

link_speed - speed in Mbps
link_duplex - 1 half duplex, 2 full duplex, 0 down

Installed Root Startup Files

```
# vi /etc/profile

if [ "$LOGNAME" = "root" ]; then
PATH=/usr/sbin:/usr/bin:/usr/local/bin:/usr/ucb
HISTFILE=/.sh_history
HISTSIZE=200
MANPATH=/usr/share/man:/usr/local/man:/opt/VRTSvmsa/man:/opt/VRTSvxvm/man
EDITOR=vi
PS1="ROOT@`/usr/ucb/hostname`# "
ENV=/.kshrc
umask 077
export PATH HISTFILE HISTSIZE MANPATH EDITOR PS1 ENV
fi
TERM=vt100
export TERM
logger -p local0.info "User $LOGNAME has logged in"
trap  2 3

# touch /.profile
# chmod 700 /.profile

# vi /.kshrc
#
# This file is read upon execution of the korn shell
# /.profile is read before this
#
HNAME=`uname -n`
PS1="$HNAME "'$PWD'" > "; export PS1

set -o vi
set -o noclobber
alias rm='rm -i'
stty erase ^h

# chmod 700 /.kshrc
```

Created the man Database

```
# catman –w
```

After this change, man –k will allow users to search for commands using keywords.

Created E-mail Aliases

```
# vi /etc/aliases
# status sends to Administrator e-mail accounts
status:jsmith@domain.com,bsmith@domain.com
# monitor sends to Administrator e-mail accounts and cell phones
monitor: jsmith@domain.com,bsmith@domain.com,6085551212@pagenet.net
# operations sends to the 24 hour operations staff
operations:operator@domain.com
```

```
# newaliases
/etc/mail/aliases: 6 aliases, longest 32 bytes, 170 bytes total
```

NOTE: By default, the scripts included within this document send notification to the status and monitor e-mail aliases.

Forwarded Mail to the Mail Server

```
# vi /etc/mail/sendmail.cf
#DSmailhost.$m
DShostname.domain.com
```

Used the fully qualified name of the mail server.

Created Root's .forward File

```
# vi /.forward
status
```

All mail is forwarded to the e-mail account specified in a .forward file. No mail remains on the server. If mail is relayed to LAN e-mail accounts, administrators and users will notice it earlier than if it remains on the server. Multiple accounts can be separated by commas.

Created Home Directories

```
# ls -ld export
drwxrwxr-x   3 root     sys          512 Aug  3 13:38 export/
# chmod 755 export
# cd /export
# mkdir home
# ls –ld /export/home
drwxr-x---   4 root     other        512 Aug  3 13:39 /export/home/
```

The rationale behind this configuration is to allow sendmail to use user's .forward files to send mail to their LAN e-mail accounts. The following section is from the sendmail man page:

"Additional restrictions have been put in place on .forward and :include: files. These files and the directory structure that they are placed in cannot be group or world-writable directories."

Created Administration Directories

```
# mkdir –p /var/adm/log/backup
# mkdir –p /var/adm/log/mon_perf
# mkdir –p /var/adm/log/perf_log
# mkdir –p /opt/admin/downloads
# mkdir –p /opt/admin/scripts/funcs
```

Created a List of Valid Shells in /etc/shells

```
# vi /etc/shells
/bin/sh
```

```
/bin/ksh
/bin/csh
/bin/bash

# chown root:other /etc/shells
# chmod 644 /etc/shells
```

If a user's shell is not included here, they may be unable to use FTP. Ensure that all shells are represented in this file.

Ensured the System Does not act as a Router

```
# touch /etc/notrouter
# chown root:sys /etc/notrouter
# chmod 444 /etc/notrouter
```

Enabled Performance Logging

```
# su – sys
# EDITOR=vi; export EDITOR
# crontab –e

# The sys crontab should be used to do performance collection. See cron
# and performance manual pages for details on startup.
#
0 * * * 0-6 /usr/lib/sa/sa1
20,40 6-22 * * 1-5 /usr/lib/sa/sa1
5 18 * * 1-5 /usr/lib/sa/sa2 -s 8:00 -e 18:01 -i 1200 -A
```

Disabled Auto Boot

```
# eeprom auto-boot?=false
```

When the server boots from a powered off state, it will stop at the OK prompt.

Configured Unique MAC Addresses

Solaris assigns the same MAC address to all NICs by default. This configuration has the potential to cause problems. (i.e. collisions and low performance). To avoid this risk, accomplish the following:

```
# eeprom local-mac-address\?=true
```

SECURITY CONFIGURATIONS

Installed SSH

Telnet and FTP pass user ids and passwords "in the clear". This sensitive information can be picked up by a sniffer. SSH encrypts traffic, effectively replacing telnet and FTP. I still recommend hardening telnet and FTP as defense in depth measures.

Commercial SSH: http://www.ssh.com
SSH Freeware: http://www.openssh.org

Restricted Root Access to the Console or su

Telnet:

```
# vi /etc/default/login

CONSOLE=/dev/console
```

Ensured that the CONSOLE entry is not commented out. To enhance accountability of administrative access, direct logon to the root account should be denied. This configuration forces users to login to their account and use the su command to access root. Root can still be accessed directly at the system console.

SSH:

```
# vi /etc/sshd_config
PermitRootLogin no

# ps -ef | grep sshd
# kill -HUP <sshd PID>
```

Restricted Access to the su Command

After these configurations, root access requires 4 elements: the user id and password of an account belonging to the group wheel and the root password.

Created the Wheel Group

```
# groupadd wheel
```

Added Administrators to the System

```
# useradd -c "John Smith" -d /export/home/jsmith -m -u 1001 -g wheel -s /bin/ksh jsmith
```

NOTE: "-g" determines the default group from /etc/group (use GID or group name)
"-u" must be a unique UID from /etc/passwd

```
# passwd jsmith (set the user's password)

# passwd -f jsmith (forced the user to change the password)

# vi /export/home/jsmith/.forward (forwards user's e-mail)
jsmith@domain.com

# chown jsmith:wheel /export/home/jsmith/.forward
```

Changed Ownership of the su Command

```
# cd /usr/bin
# ls -al su
-r-sr-xr-x   1 root     sys        17976 Oct  6  1998 su
# /usr/bin/chgrp wheel su
# /usr/bin/chmod 4750 su
# ls -al su
-rwsr-x---   1 root     wheel      17976 Oct  6  1998 su
# cd /sbin
# ls -al su.static
-r-xr-xr-x   1 root     sys       473808 Sep  1  1998 su.static
# /usr/bin/chgrp wheel su.static
# /usr/bin/chmod 4750 su.static
# ls -al su.static
-rwsr-x---   1 root     wheel     473808 Sep  1  1998 su.static
```

* From Lance Spitzer's Armoring Solaris

Set the Password Policy

```
# vi /etc/default/passwd
```

Before:
MAXWEEKS=
MINWEEKS=
PASSLENGTH=6

After:
MAXWEEKS=8
MINWEEKS=1
PASSLENGTH=8
WARNWEEKS=1

Root and user passwords are set to expire at the 3 month mark. If the root password expires, it must be reset from the system console. To avoid lockout, reset the root passwords at the 2 month mark.

Definitions:

MAXWEEKS - Maximum time period that a password is valid.

MINWEEKS - Minimum time period before a password can be changed.

PASSLENGTH - Minimum length of a password, in characters.

WARNWEEKS - Time period until warning of date of password's ensuing expiration.

Configured Disconnect After 3 Login Failures

```
# vi /etc/default/login
```

```
# Disconnect users after three login failures
```

```
RETRIES=3
```

NOTE: By default, Solaris will terminate a connection after 5 consecutive login failures. Set retries to 3. This is an industry standard (e.g. 3 strikes you're out).

```
# The SYSLOG_FAILED_LOGINS variable is used to determine how many failed
# login attempts will be allowed by the system before a failed login
# message is logged, using the syslog(3) LOG_NOTICE facility.  For example,
# if the variable is set to 0, login will log -all- failed login attempts.
#
SYSLOG_FAILED_LOGINS=3
```

Disabled the rlogin Command

Commented out the following lines in /etc/pam.conf:

```
#rlogin  auth sufficient /usr/lib/security/pam_rhosts_auth.so.1
#rlogin  auth required /usr/lib/security/pam_unix.so.1
#rsh     auth required /usr/lib/security/pam_rhosts_auth.so.1
```

This configuration forces users to use their passwords with the rlogin command.

Locked Down Remote Access Files

```
# /usr/bin/touch /.rhosts /.netrc /etc/hosts.equiv
# /usr/bin/chmod 0 /.rhosts /.netrc /etc/hosts.equiv
```

* From Lance Spitzer's Armoring Solaris

These files provide "trusted" users remote access without the use of passwords. An alternative would be to ensure that they do not exist and use monitoring software to notify if they are created.

Removed or Disabled Unnecessary Accounts

```
# passwd –l adm
# passwd –l bin
# passwd –l daemon
# passwd –l listen
# passwd –l lp
# passwd –l nobody
# passwd –l noaccess
# passwd –l nuucp
# passwd –l sys
# passwd –l uucp
```

The nobody4 account is no longer needed.

```
# userdel nobody4
```

Assigned Disabled Accounts an Invalid Shell

```
# vi /sbin/noshell
#!/bin/sh
#
# Solaris 2.X Disabled Account Access Script
# Purpose: Sends notification when someone attempts
# to access an account that has been disabled.
# Usage: Save as /sbin/noshell. Use as the shell in
# /etc/passwd for accounts that have been disabled.
# Dependencies: None
# Outputs: e-mail and syslog
# Author: Unknown (perhaps originating from Titan scripts)
# Modifications: Added notification via e-mail – gtr
#*************************************************************
#:
trap "" 1 2 3 4 5 6 7 8 9 10 12 15 19

HOSTNAME=`uname -n`
USER=`id | awk '{print \$1}'`
logger -i -p auth.err "Attempted access by $USER on host $HOSTNAME"

# The next variable can be set for multiple addresses
# (i.e. jsmith@yahoo.com,jsmith@hotmail.com)
MAILADD=monitor

mail $MAILADD <<EOF
From: $0
To: $MAILADD
Subject: Unauthorized Access Attempt on $HOSTNAME
Someone has attempted to access a disabled account ($USER)
on $HOSTNAME. Please investigate immediately.

$DATE

EOF

echo "Sorry"
wait
exit 0

# chmod 755 /sbin/noshell

# vi /etc/passwd
daemon:x:1:1::/:/usr/sbin/noshell
```

Assign the shell /sbin/noshell as the shell for accounts that should never be allowed to log in (i.e. daemon, bin, sys, adm, lp, smtp, uucp, nuucp, listen, nobody, and noaccess).

As an alternative, the noshell binary can be used (http://www.cert.org/security-improvement/implementations/i049.02.html). When compared to the script, its benefit is that it is compiled code. Its downside is that administrators do not receive e-mail notification.

Restricted FTP Usage

Ensured /etc/ftpusers contained the following accounts:

```
# vi /etc/ftpusers
root
adm
bin
daemon
listen
lp
nobody
noaccess
nobody4
nuucp
smtp
sys
uucp
```

These system accounts no longer have the ability to FTP into the server. Any additional administrative accounts should be added as well (i.e. oracle, webadmin, etc).

Secured the IP Module

Downloaded the latest nddconfig script from: http://wwws.sun.com/blueprints/tools/nddconfig_license.html

```
# vi /etc/init.d/nddconfig
# chmod 740 /etc/init.d/nddconfig
# chown root:sys /etc/init.d/nddconfig
# ln /etc/init.d/nddconfig /etc/rc2.d/S70nddconfig
```

Randomized the Initial Sequence Number of all TCP Connections

Configured on-line:

```
# ndd -set /dev/tcp tcp_strong_iss 2
```

Configured for reboot:

```
# vi /etc/default/inetinit
TCP_STRONG_ISS=2
```

Randomizing the initial sequence number of TCP connections protects the system against session hijacking and IP spoofing.

* From Lance Spitzer's Armoring Solaris

Disabled Unnecessary Services in /etc/inetd.conf

```
# vi /etc/inetd.conf
# ps –ef | grep inetd
# kill –HUP <inetd PID>
```

Commented out all entries including telnet and ftp. Used SSH and scp as replacements. They offer additional security. Many of these unnecessary services contain easily exploitable security vulnerabilities. Be advised, some programs add entries to the end of inetd.conf and cannot function without them (Solstice Disk Suite for example).

Disabled Start Scripts

In general, disable any unnecessary services. This will address security vulnerabilities and, in some cases, increase performance. Ensure that you understand the purpose of a service before disabling it. Listed below are the services I typically disable. With new versions of Solaris, there may be more to consider.

Disabled Volume Management

```
# cd /etc/rc2.d
# mv S92volmgt s92volmgt
```

After this configuration, CD-ROMs will not be automatically mounted. To manually mount a CD-ROM use:

```
# mount -F hsfs -o ro /dev/dsk/c0t6d0s0 /mnt
```

Disabled Dtlogin

Dtlogin is disabled if the server is not intended to run the Common Desktop Environment (CDE) or GUIs.

```
# cd /etc/rc2.d
# mv S99dtlogin s99dtlogin
```

Disabled Printing

```
# /usr/lib/lpshut
# cd /etc/rc2.d
# mv S80lp s80lp
```

Disabled RPC

RPC is disabled if the server is not intended to run CDE. To determine what is using rcp, use "`rpcinfo -p`".

```
# cd /etc/rc2.d
# mv /etc/rc2.d/S71rpc /etc/rc2.d/s71rpc
```

Disabled the NFS Client

```
# /etc/init.d/nfs.client stop
# cd /etc/rc2.d
# mv S73nfs.client s73nfs.client
```

Disabled the NFS Server

```
# /etc/init.d/nfs.server stop
# cd /etc/rc3.d
# mv S15nfs.server s15nfs.server
```

Disabled UUCP

```
# cd /etc/rc2.d
# mv S70uucp s70uucp
```

Disabled the LDAP Client

```
# cd /etc/rc2.d
# mv S71ldap.client s71ldap.client
```

Disabled the Auto Mounter

```
# /etc/init.d/autofs stop
# cd /etc/rc2.d
# mv S74autofs s74autofs
```

Disabled the Network Time Daemon

```
# /etc/init.d/xntpd stop
# cd /etc/rc2.d
# mv S74xntpd s74xntpd
```

Disabled the Logical Link Control Driver

```
# cd /etc/rc2.d
# ./S40llc2 stop
# mv S40llc2 s40llc2
```

Disabled Auto Install

```
# cd /etc/rc2.d
# mv S72autoinstall s72autoinstall
```

Disabled Cachefs Daemon

```
# cd /etc/rc2.d
# mv S73cachefs.daemon s73cachefs.daemon
```

Disabled Asynchronous PPP Daemon

```
# cd /etc/rc2.d
# mv S47pppd s47pppd
```

Disabled cacheos.finish Script

```
# cd /etc/rc2.d
# mv S93cacheos.finish s93cacheos.finish
```

Disabled Preservation of Files Killed by Vi

```
# cd /etc/rc2.d
# mv S80PRESERVE s80PRESERVE
```

Disabled Power Management

```
# cd /etc/rc2.d
# mv S85power s85power
```

Disabled Flash Prom Update

```
# cd /etc/rc2.d
# mv S75flashprom s75flashprom
```

Before attempting to update the eeprom, temporally enable this script.

Disabled "Buttons n Dials-Setup"

```
# cd /etc/rc2.d
# mv S89bdconfig s89bdconfig
```

Disabled Spc

```
# cd /etc/rc2.d
# mv S80spc s80spc
```

Disabled Sun Management Center

```
# cd /etc/rc2.d
# mv S90wbem s90wbem
```

Disabled Network Cache and Accelerator

```
# cd /etc/rc2.d
# mv S94ncalogd s94ncalogd
# mv S95ncad s95ncad
```

Used to increase web server performance

Disabled Mobile IP Agent

```
# cd /etc/rc3.d
# mv S80mipagent s80mipagent
```

Disabled SNMP

```
# cd /etc/rc3.d
# /usr/bin/pkill -9 -x -u 0 '(snmpdx|snmpv2d|mibiisa)'
# mv S76snmpdx s76snmpdx
```

Disabled Apache

```
# cd /etc/rc3.d
# mv S50apache s50apache
```

Disabled DMI

```
# cd /etc/rc3.d
# /usr/bin/pkill -9 -x -u 0 '(snmpXdmid|dmispd)'
# mv S77dmi s77dmi
```

Disabled the Sendmail Daemon

The system continues to send mail out. It does not receive mail in to the server. This eliminates a significant security vulnerability.

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

Prevented sendmail from starting at boot:

```
# cd /etc/rc2.d
# mv S88sendmail s88sendmail
```

Ensured the sendmail queue is cleaned out:

```
# crontab –e

# The Sendmail daemon is not running - This tells it to send mail out
05,20,35,50 * * * * /usr/lib/sendmail –q
```

Disabled Multicasting

Multicasting is typically used for clustering. Ensure that it is not required by an application.

```
# vi /etc/init.d/inetsvc

#
# Add a static route for multicast packets out our default interface.
# The default interface is the interface that corresponds to the node name.
```

```
#
#mcastif=`/sbin/dhcpinfo Yiaddr`
#
#if [ $? -ne 0 ]; then
#       mcastif=`uname -n`
#fi
#
#echo "Setting default interface for multicast: \c"
#/usr/sbin/route add -interface -netmask "240.0.0.0" "224.0.0.0" "$mcastif"
```

Disabled the Serial Port Listeners

This configuration can be accomplished unless there is a modem or console terminal attached to the system.

```
# vi /etc/inittab
```

Remove the line with “/usr/lib/saf/sac -t 300”

```
# chown root:sys /etc/inittab
# chmod 644 /etc/inittab
```

Added Warning Banners

These configurations replace the operating system version with a warning banner displayed during the login process.

Login:

vi /etc/motd (replaced operating system version with a warning banner)

```
Property of Company

WARNING: To protect systems from unauthorized use and to ensure that the
system is functioning properly, activities on this system are monitored and
recorded and subject to audit. Use of this system is expressed consent to such
monitoring and recording. Any unauthorized access or use of this system is
prohibited and could be subject to criminal and civil penalties.
```

```
# cp /etc/motd /etc/issue
```

Telnet:

```
# vi /etc/default/telnetd
UMASK=022
BANNER=""
# chown root:sys /etc/default/telnetd
# chmod 444 /etc/default/telnetd
```

FTP:

```
# vi /etc/default/ftpd
UMASK=022
BANNER=`cat /etc/motd`
# chown root:sys /etc/default/ftpd
# chmod 444 /etc/default/ftpd
```

Defined PATH, SUPATH and UMASK in /etc/default/login

```
# vi /etc/default/login

PATH=/usr/sbin:/usr/bin
SUPATH=/usr/sbin:/usr/bin
UMASK=027
```

Disabled World Access in Default Umask

Added "umask 027" to the following files:

```
/etc/profile (change)
/etc/.login (add)
/etc/skel/local.profile (add)
/etc/skel/local.login (add)
/etc/skel/local.cshrc (change)
```

Ensured no Alternate UID 0 Accounts Exist

```
# more /etc/passwd
```

Ensure that root is the only account with a UID of 0 in the 3rd field of the /etc/password file. UID 0 identifies an account as root to the operating system. Any alternate account with a UID of 0 is given /usr/sbin/noshell as a login shell.

Ensured all Accounts have Passwords

```
# logins -p
```

Use the command logins -p to check for accounts that do not require a password to log in.

Restricted Access to the "at" and "crontab" Commands

These accesses should be given out on an as needed basis.

Determine who has a crontab file:

```
# ls /var/spool/cron/crontabs
```

Restrict the use of "at" and "crontab". Only users listed in these files will be allowed to use "at" and "crontab". Start with the root user. Add sys for performance logging and lp for print queue maintenance:

```
# vi /etc/cron.d/cron.allow
# chmod 600 /etc/cron.d/cron.allow
# cp –p /etc/cron.d/cron.allow /etc/cron.d/at.allow
```

Create an /etc/cron.d/cron.deny file. Users listed in this file will not have access to “at” and “crontab”:

```
# cat /etc/passwd | cut -f1 -d: | grep -v root >> /etc/cron.d/cron.deny
# chmod 600 /etc/cron.d/cron.deny
```

Create an /etc/cron.d/at.deny file:

```
# cp -p /etc/cron.d/cron.deny /etc/cron.d/at.deny
```

Replicated Syslog to the Monitoring Console

Replicating syslog to a central system makes it difficult for an intruder to entirely hide their tracks. As syslog entries are created locally, they are immediately copied to the central syslog server. Daily review of the centralized logs is also an effective way to detect system anomalies (i.e. hardware failures, software errors, etc).

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

# vi /etc/hosts
```

Before:

```
192.168.1.101     sunsrv01.domain.com sunsrv01 loghost
```

After:

```
192.168.1.101     sunsrv01.domain.com sunsrv01
192.168.1.102     sunsrv02 loghost

# cp /etc/syslog.conf /etc/syslog.conf.orig
# vi /etc/syslog.conf

# next 2 lines added for syslog replication
*.err;kern.notice;auth.notice;user.none                 @loghost
*.err;kern.debug;daemon.notice;mail.crit;user.none      @loghost
```

NOTE: The entries must be separated by tabs.

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

Forwarded Root Access Attempts to the System Console

```
# vi /etc/default/su
```

`CONSOLE=/dev/console` (uncommented)

Enabled Logging of the su Command

This configuration logs both success and failure of su command usage.
NOTE: This configuration is required by the root login notification script (below).

```
# vi /etc/default/su
```

SULOG=/var/adm/sulog (uncommented)

```
# cd /var/adm
# touch sulog
# chgrp sys sulog
# chmod 600 sulog
```

Enabled AUTH Logging

The auth facility controls account access with login, su, etc.

```
# vi /etc/syslog.conf
```

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

NOTE: The entries must be separated by tabs.

```
# /etc/init.d/syslog stop
# /etc/init.d/syslog start
```

Enabled Logging of Unsuccessful Login Attempts

The loginlog file records consecutive failed login attempts.

```
# cd /var/adm
# touch loginlog
# chgrp sys loginlog
# chmod 600 loginlog
```

Enabled Logging of Successful Logins

```
# cd /var/log
# touch logins
# chgrp sys logins
# chmod 600 logins

# vi /etc/syslog.conf

# log successful logins
local0.info                                             /var/log/logins
```

NOTE: The entries must be separated by tabs.

```
# /etc/init.d/syslog stop
# /etc/init.d/syslog start
```

Added the following entry to /etc/profile and /etc/.login:

```
logger -p local0.info "User $LOGNAME has logged in"
```

Enabled Logging of CDE Login Attempts

```
# vi /etc/pam.conf
```

Added the word “debug” after the account management entries

```
#
# Account management
#
login   account required        /usr/lib/security/$ISA/pam_unix.so.1 debug
dtlogin account required        /usr/lib/security/$ISA/pam_unix.so.1 debug
```

```
# vi /etc/syslog.conf
```

Added “;auth.debug;user.debug” to the line that logs successful logins

```
# log successful logins
local0.info;auth.debug;user.debug               /var/log/logins
```

NOTE: The entries must be separated by tabs.

```
# /etc/init.d/syslog stop
# /etc/init.d/syslog start
```

Log Incoming Connections for TCP Services

```
# vi /etc/syslog.conf
```

```
# log incoming connections for TCP services
daemon.notice                                   /var/log/syslog
```

NOTE: The entries must be separated by tabs.

```
# /etc/init.d/syslog stop
# /etc/init.d/syslog start
```

```
# vi /etc/rc2.d/S72inetsvc
```

(change the following entry:)

```
     /usr/sbin/inetd -s
```

(to read:)

```
     /usr/sbin/inetd -s -t
```

Enabled Auditing

Solaris provides the Basic Security Module (BSM) to audit actions taken by users. There is a relatively small performance hit associated with its use. BSM provides forensic evidence. For more detail, see Sun’s article on BSM (http://www.sun.com/solutions/blueprints/0201/audit_config.pdf).

Enabled BSM

```
# /etc/security/bsmconv
# /usr/sbin/shutdown –i6 –g0 –y
```

Configured the Classes of Events to Log

```
# vi /etc/security/audit_control
dir:/var/audit
flags:lo,ad,pc,fc,fd,fm
naflags:lo,ad
#
#   lo - login/logout events
#   ad - administrative actions: mount, exportfs, etc.
#   pc - process operations: fork, exec, exit, etc.
#   fc - file creation
#   fd - file deletion
#   fm - change of object attributes: chown, flock, etc.
#
```

Audit all Actions Taken by Root

```
# vi /etc/security/audit_user
# log all of the commands that the root user runs
root:lo,ex:
```

Installed a Log Rotation Script

```
# touch /etc/security/newauditlog.sh
# chmod 700 /etc/security/newauditlog.sh
# mkdir -p /var/audit/logs
# vi /etc/security/newauditlog.sh
#!/bin/ksh
#
# Solaris Basic Security Module (BSM) Log Rotation Script
# newauditlog.sh - Start a new audit file and expire the old logs
#
# Source: Solaris Security Guide
# Modifications: Added log compression and deletion with e-mail
# notification when the log directory grows past a certain size.
# - gtr
#
#****************************************************************

PATH=/usr/sbin:/usr/bin
AUDIT_EXPIRE=30
AUDIT_DIR=/var/audit
LOG_DIR=/var/audit/logs

# Rotate the audit log
```

```
/usr/sbin/audit -n

# Move log files to the archive directory and compress

for i in `/usr/bin/ls $AUDIT_DIR | grep -v not_terminated | grep -v logs`
do

compress $AUDIT_DIR/$i
mv $AUDIT_DIR/$i.Z $LOG_DIR/$i.Z

done

# Delete old log files

cd $LOG_DIR # in case it is a link
/usr/bin/find . $LOG_DIR -type f -mtime +$AUDIT_EXPIRE -exec rm {} > /dev/null 2>&1 \;

# Ensure that log files do not take up more than 250MB

# The next variable can be set for multiple addresses
# (i.e. jsmith@yahoo.com,jsmith@hotmail.com)
MAILADD=status

# The maximum size $OUTPUTDIR is allowed to reach before log files
# are deleted. (250000=250MB)
MAXSIZ=250000

LOGDU=`du -sk $LOG_DIR | awk '{ print $1 }`

if [ "$LOGDU" -gt "$MAXSIZ" ]; then
           find $LOG_DIR -mtime +21 -exec rm {} \;
           mail $MAILADD <<EOF
From: $0
To: $MAILADD
Subject: Security Audit Log Size on `uname -n`
$LOG_DIR was $LOGDU KB. $0 does not
allow more than 250 MB of log files in this directory.
Log files older than 21 days have been deleted.
The current size of $LOG_DIR is `du -sk $LOG_DIR | awk '{ print $1 }` KB.
Thank you.
EOF
fi

exit 0
```

Run the Script Nightly from Cron

```
# EDITOR=vi; export EDITOR
# crontab –e
0 0 * * * /etc/security/newauditlog.sh
```

NOTE: Use the praudit command to convert audit data into ASCII format:

```
# cd /var/audit
# praudit logfile
```

* From the Solaris Security Guide

INSTALLED MONITORING SCRIPTS

Root Login Notification Script (rtlgn.sh)

Purpose: Monitors root logins via the su command and directly at the console. Notifies via e-mail.
Dependencies: /var/adm/sulog
/etc/aliases – status (e-mail addresses of administrators)

```
# vi /opt/admin/scripts/rtlgn.sh
#!/bin/ksh
#
# Solaris 2.X Root Login Notification Script
# Purpose: Sends notification when root logs in
# Usage: Execute from crontab every 15 minutes
# 14,29,44,59 * * * * /opt/admin/scripts/rtlgn.sh > /dev/null
# Dependencies: None
# Outputs: E-mail
#*************************************************************

PATH=/usr/bin:/usr/sbin:/usr/ucb:/bin
SRVNM=`uname -n`
DATE=`date '+%m/%d'`
DAY=`date '+%d'`
HOUR=`date '+%H'`
MONTH=`date '+%m'`
MIN=`date '+%M'`

LOGDIR=/var/adm/log/rtlgn
DATFILE=$LOGDIR/rtlgn.dat

if [ !  -d $LOGDIR ] ; then
      mkdir -p $LOGDIR
      touch $DATFILE
fi

# Clean out the dat file each day

if [ $HOUR -eq "00" ]; then

 if [ $MIN -lt "15" ]; then
        > $DATFILE
 fi

fi


# The next variable can be set for multiple addresses
# (i.e. jsmith@yahoo.com,jsmith@hotmail.com)
MAILADD=monitor
```

```
# Check for remote root login (should never happen)
#who

# Check for recent root console login
# Determine if notification has been sent this hour
if [ `grep -c "$DATE $HOUR CONSOLE" $DATFILE` -eq 0 ]
then

if [ `last root console | grep -c "$MONTH  $DAY $HOUR" ` -gt 0 ]
then

mail $MAILADD <<EOF
From: $0
To: $MAILADD
Subject: Root Console Login $SRVNM

A root console login has occurred:

`last root console | grep "$MONTH  $DAY $HOUR"`

EOF

# Ensure notification only occurs once per hour
print "$DATE $HOUR CONSOLE" >> $DATFILE

fi
fi

# Check for recent su to root
# Determine if notification has been sent this hour
if [ `grep -c "$DATE $HOUR SU" $DATFILE ` -lt 1 ]
then

if [ `grep "$DATE $HOUR" /var/adm/sulog | grep -v root- | grep root | grep -c "+" ` -gt 0
]
then

mail $MAILADD <<EOF
From: $0
To: $MAILADD
Subject: Root Access on $SRVNM

The following root login has occurred:

`grep "$DATE $HOUR" /var/adm/sulog | grep root | grep "+"`

EOF

# Ensure notification only occurs once per hour
print "$DATE $HOUR SU" >> $DATFILE

fi
fi

exit 0
```

```
# chmod 700 /opt/admin/scripts/rtlgn.sh
```

System Boot Notification Script (S99notify)

Purpose: Sends notification when a server boots.
Dependencies: None
/etc/aliases – monitor (administrators' e-mail and pagers)

```
# vi /etc/rc2.d/S99notify
#!/bin/ksh
#
# Solaris 2.X Boot Notification Script
# /etc/rc2.d/S99notify - Sends e-mail notification to administrators
# when the system is booted.
#
#*****************************************************************

PATH=/usr/sbin:/usr/bin

SRVNM=`uname -n`

# The next variable can be set for multiple addresses
# (i.e. jsmith@yahoo.com,jsmith@hotmail.com)
MAILADD=monitor

mail $MAILADD <<EOF
From: $0
To: $MAILADD
Subject: Boot of $SRVNM

$SRVNM has booted up.

If this is news to you, please investigate.

`date`

EOF

exit 0
```

```
# chmod 700 /etc/rc2.d/S99notify
```

Installed LogSentry

LogSentry parses /var/adm/messages and sends notification based on the hacking and violation files. Customize the ignore file to reduce false positives. Execute from cron to send a report once per day, with notification sent to administrator's e-mail accounts. It makes sense to centralize syslog to a single server and run LogSentry there.

http://www.psionic.com/products/logsentry.html

File System Monitoring Script (mon_fs.sh)

Purpose: Monitors the size of file systems. Notifies via e-mail.

Dependencies: mon_fs.dat – Contains which file systems to monitor and how large they can be before a warning is issued.
/etc/aliases – status (e-mail addresses of administrators)

```
# vi /opt/admin/scripts/mon_fs.sh
#!/bin/ksh
#
# Solaris 2.X Monitor File Systems Script
# Purpose: Check to see if file systems are filling up
# Usage: Execute from crontab
# Dependencies: mon_fs.dat
# Outputs: E-mail
#*************************************************

# The directory this script resides in
ADMINDIR=/opt/admin/scripts

# The next variable can be set for multiple addresses
# (i.e. jsmith@yahoo.com,jsmith@hotmail.com)
MAILADD=monitor

# Define the hostname of the server
SRVNM=`uname -n`

while read -r FS MAXCAP
do

CAPACITY=`df -k $FS | grep -v avail | awk {'print $5'} | awk -F% {'print
$1'}`

if test $CAPACITY -gt $MAXCAP; then
           mail $MAILADD <<EOF
From: $0
To: $MAILADD
Subject: File System on $SRVNM
$FS is at $CAPACITY% capacity on $SRVNM (Threshold is $MAXCAP%).

`date`

EOF
fi

done < $ADMINDIR/mon_fs.dat

exit 0


# vi /opt/admin/scripts/mon_fs.dat
/     90
/var  90
/opt  90

# chmod 600 /opt/admin/scripts/mon_fs.dat
# chmod 700 /opt/admin/scripts/mon_fs.sh
```

Process Monitoring Script (mon_procs.sh)

Purpose: Ensures processes are running. Notifies via e-mail.
Dependencies: mon_procs.dat – Contains the names of processes
/etc/aliases – status (e-mail addresses of administrators)

```
# vi /opt/admin/scripts/mon_procs.sh
#!/bin/ksh
#
# Solaris 2.X Monitor Processes Script
# Purpose: Check to see if processes are running
# Usage: Execute from crontab
# Dependencies: mon_procs.dat
# Outputs: E-mail
#************************************************

# The directory this script resides in
ADMINDIR=/opt/admin/scripts

# The next variable can be set for multiple addresses
# (i.e. jsmith@yahoo.com,jsmith@hotmail.com)
MAILADD=monitor

SRVNM=`uname -n`

while read PROG
do
ANSWER=`ps -e -o comm | grep $PROG`
if test "$ANSWER" = "$PROG"; then
            sleep 1
else
            mail $MAILADD <<EOF
From: $0
To: $MAILADD
Subject: Missing process on $SRVNM
Checking $PROG on $SRVNM... not found!

EOF
fi
done < $ADMINDIR/mon_procs.dat

exit 0


# vi /opt/admin/scripts/mon_proc.dat
/usr/sbin/syslogd

# chmod 700 /opt/admin/scripts/mon_procs.sh
# chmod 600 /opt/admin/scripts/mon_procs.dat
```

Server Monitoring Script (mon_srv.sh)

Purpose: Ensures servers respond to ping. Notifies via e-mail.

Dependencies: mon_srv.dat – Contains IP addresses, monitor e-mail address, and server names
/etc/aliases – monitor (administrators' e-mail and pagers)

```
# vi /opt/admin/scripts/mon_srv.sh
#!/bin/ksh
#
# Solaris 2.X Monitor Servers Script
# Purpose: Monitors servers with the ping command
# and notifies via e-mail.
# Usage: Execute from crontab
# Dependencies: /opt/admin/scripts/mon_srv.dat
# Outputs: E-mail
#************************************************

# The directory this script resides in
ADMINDIR=/opt/admin/scripts

# The next variable can be set for multiple addresses
# (i.e. jsmith@yahoo.com,jsmith@hotmail.com)
MAILADD=monitor

while read -r IP SRVNM
do
    if test `/usr/sbin/ping $IP | grep -c "is alive"` -eq 0; then
        # Wait 5 minutes before checking again
      sleep 300
      if test `/usr/sbin/ping $IP | grep -c "is alive"` -eq 0; then

           mail $MAILADD <<EOF
From: $0
To: $MAILADD
Subject: $SRVNM Down

$SRVNM is not responding.

EOF
      fi
    fi

done < $ADMINDIR/mon_srv.dat
exit 0


# vi /opt/admin/scripts/mon_srv.dat
192.168.1.103     hostname

# chmod 700 /opt/admin/scripts/mon_srv.sh
# chmod 600 /opt/admin/scripts/mon_srv.dat
```

User Disk Space Monitoring Script (maildu.sh)

Purpose: Notifies users when their home directory reaches over 100 MB. Sends e-mail to LAN accounts.
Dependencies: ~/.forward – Contains LAN e-mail addresses of users.

```
# vi /opt/admin/scripts/maildu.sh
```

```
#!/bin/ksh
#
# Solaris 2.X Mail Disk Usage Script
# Purpose: Notifies users via e-mail when their home
# directories contain more than 100 MB of files
# Usage: Run this script from crontab. Do not send
# the output to /dev/null. The only output it
# produces is which directories are too large.
# Dependencies: None
# Outputs: E-mail
#************************************************

PATH=/usr/sbin:/usr/bin:/usr/ucb:/bin:.

# Where the user's home directories reside
HOMEDIR=/export/home

# Define the hostname of the server
SRVNM=`uname -n`

# Ensure that temp files get cleaned up upon exit
trap '/bin/rm -fr $tmp; exit' 0 1 2 3 15
WRKFILE=/tmp/prog$$

# Checks space used by users

cd $HOMEDIR
du -sk * | sort -nr >> $WRKFILE

# Notifies users

while read -r MB NAME
do

# 1 MB = 1024 KB

if [ "$MB" -gt "102400" ]; then
           # Notify the root user
           print "Mailing Disk Usage reminders out to:\n"
           print " $NAME \t$MB KB\n"
           # Notify the user
                if [ -f $HOMEDIR/$NAME/.forward ]
                then
                    MAILADD=`cat $HOMEDIR/$NAME/.forward`
                else
                    MAILADD=$NAME
                fi
           mail $MAILADD <<EOF
From: $0
To: $MAILADD
Subject: Disk Usage on $SRVNM
The automated disk usage utility indicates that you
have $MB KB's of disk usage in your home directory on
$SRVNM. You receive mail if you have more than 100 MB
in your home directory.  Please delete any excess
files you may have. Thank you.
```

```
UNIX System Administrators
EOF

fi

done < $WRKFILE

rm $WRKFILE

exit 0
```

chmod 700 maildu.sh

Performance Monitoring Script (mon_prf.sh)

Purpose: Monitors the performance of the server. Uses vmstat, iostat, netstat, and other performance commands. Notifies via e-mail.
Dependencies: /etc/aliases – status (e-mail addresses of administrators)

vi /opt/admin/scripts/mon_prf.sh

```
#!/bin/ksh
#
# Solaris 2.X Performance Monitoring Script
# Purpose: This script executes performance commands and notifies via
#         e-mail when performance is poor.
# Useage: Execute the script from crontab at 30 minute intervals.
# Dependencies: None
# Outputs: Logfiles and e-mail
# CRONTAB EXAMPLE (Mon - Fri 7am - 6pm):
# 17,47 7-18 * * 1-5 /opt/admin/scripts/mon_prf.sh
#
#**************************************************************************
PATH=$PATH:/usr/sbin:/usr/bin


# Define the server's hostname
SRVNM=`uname -n`

# The directory this script resides in
ADMINDIR=/opt/admin/scripts

# Create log directory
DATDIR=/var/adm/log/mon_perf

if [ !  -d $DATDIR ] ; then

      mkdir -p $DATDIR
fi

# The next variable can be set for multiple addresses
# (i.e. jsmith@yahoo.com,jsmith@hotmail.com)
MAILADD=monitor

# vmstat
VMDAT=`vmstat 1 2 | tail -1`
```

```
# CPU Performance (vmstat - r column)
# When the 'r' or run queue column reaches above 3 processes per CPU,
# there is insufficient CPU power, and jobs are spending an
# increasing amount of time in the queue before being assigned to a CPU.
# This reduces throughput and increases interactive response time.

CPUPERF=`echo $VMDAT | awk '{ print $1 }`
    if [ "$CPUPERF" -gt "3" ]; then
            mail $MAILADD <<EOF
From: $0
To: $MAILADD
Subject: CPU Performance on $SRVNM
The vmstat run queue column has exceeded 3 processes per CPU on
$SRVNM. There is insufficient CPU power for the load placed on
the server.
EOF

fi

# CPU Performance (vmstat - cpu id column)
# The cpu id column indicates what % the cpu is idle

CPUSTAT=`echo $VMDAT | awk '{ print $22 }`
    if [ "$CPUSTAT" -lt "10" ]; then
            mail $MAILADD <<EOF
From: $0
To: $MAILADD
Subject: CPU Performance on $SRVNM
The vmstat cpu id column is less than 10 on $SRVNM.
The CPU is idle $CPUSTAT % of the time.
EOF

fi

# Memory Performance (vmstat - sr column)
# The experts say that when the 'sr' or scan rate column reaches above 200,
# the system is scanning through memory looking for pages to free at a high
# rate. This indicates active pages might be stolen from processes. A high
# scan rate can cause your system to consume more cpu resources than it
# normally would.

MEMSTAT=`echo $VMDAT | tail -1 | awk '{ print $12 }`

    if [ "$MEMSTAT" -gt "200" ]; then
            mail $MAILADD <<EOF
From: $0
To: $MAILADD
Subject: Memory Performance on $SRVNM
According to vmstat, the scanrate on $SRVNM is
$MEMSTAT. This indicates that there is not enough
memory to meet the server's current load.
EOF

fi

# TCP Connections
```

```
#

TCPCON=`netstat -aP tcp | tail +39 | wc -l`

    if [ "$TCPCON" -gt "900" ]; then
           mail $MAILADD <<EOF
From: $0
To: $MAILADD
Subject: TCP Connections on $SRVNM
According to netstat -a, there are currently $TCPCON
TCP connections on $SRVNM. This may or may not be
cause for concern.

EOF

fi

# NIC Input Errors
#

# netstat -i
#NETIDAT=`netstat -i | grep hme0`

#NICIE=`echo $NETIDAT | awk {'print $6'}`

#    if [ "$NICIE" -gt "10" ]; then
#           mail $MAILADD <<EOF
#From: $0
#To: $MAILADD
#Subject: NIC Input Errors on $SRVNM
#According to netstat -i, there are currently $NICIE
#input errors on $SRVNM hme0 NIC.

#EOF

#fi

# NIC Output Errors
#

#NICOE=`echo $NETIDAT | awk {'print $8'}`

#    if [ "$NICOE" -gt "10" ]; then
#           mail $MAILADD <<EOF
#From: $0
#To: $MAILADD
#Subject: NIC Output Errors on $SRVNM
#According to netstat -i, there are currently $NICOE
#output errors on $SRVNM hme0 NIC.

#EOF

#fi

# iostat
# Disk performance
# Create iostat data file
```

```
#DATFILE=$DATDIR/iostat.dat
#cp $DATFILE $DATFILE.old
#cp /dev/null $DATFILE

# Checking the iostat util column
#IODAT=`iostat -Dl 20 -n | tail -1`

#DSKSTAT=`echo $IODAT | awk '{ print $3, $6, $9, $12, $15, $18, $21, $24, $27, $30, $33,
$36, $39, $42, $45, $48, $51, $54, $57, $60 }`

# Will need to edit the next line to resolve this problem
#echo $DSKSTAT >> $DATFILE
#cat iostat.dat | awk {'print $2'}

#while read -r
#do
#        if [ "$REPLY" -gt "3" ]; then
#           mail $MAILADD <<EOF
#From: $0
#To: $MAILADD
#Subject: Disk Performance on $SRVNM
#According to iostat, the disk utilization on $SRVNM is
#greater than 3 on one of the server's hard disks. This
#indicates that the disk is being heavily used.
#EOF
#fi
#done < $DATFILE

# netstat


# CPU Data
#mpstat

# swap
#swap -l

# /tmp (Running out of swap space)
# du -sk /tmp

exit 0
```

chmod 700 /opt/admin/scripts/mon_prf.sh

Veritas Cluster Failure Notification Script (resfault)

vi /opt/VRTSvcs/bin/triggers/resfault

```
#!/bin/ksh
#
# Veritas Cluster Notification Script
# /opt/VRTSvcs/bin/triggers/resfault - The cluster triggers this when there is an
# issue with one of its managed resources.
# Inputs: <System name> <Resource>
```

```
#*************************************************************

PATH=/usr/sbin:/usr/bin

DATE="`date`"

# The next variable can be set for multiple addresses
# (i.e. jsmith@yahoo.com,jsmith@hotmail.com)
MAILADD=monitor

mail $MAILADD <<EOF
From: $0
To: $MAILADD
Subject: VCS Oracle Database Warning

$DATE

Resource $2 has faulted on system $1

If this is news to you, please investigate.

EOF

exit 0
```

chmod 700 /opt/VRTSvcs/bin/triggers/resfault

INSTALLED REPORTING / LOGGING SCRIPTS

Monitoring scripts should be used in conjunction with commercial monitoring software to provide layered monitoring (defense in depth).

System Status Script (status.sh)

Purpose: Produces a report with uptime, disk space, current and past logins, who has used the su command, interface and network configurations, and the current processes running.
Dependencies: hrdwspecs.sh
/etc/aliases – status (e-mail addresses of administrators)

vi /opt/admin/scripts/status.sh

```
#!/bin/ksh
#
# Solaris 2.X System Status Script
# Purpose: Produces a report
# Usage: Execute from the command line or crontab.
#        Save 30 days for history.
# Dependencies: $SCRIPTS/hrdwspecs.sh
# Outputs: Standard output or e-mail
# Crontab Example: # 31 7 * * 1-5 /opt/admin/scripts/status.sh jsmith@yahoo.com
#
#*****************************************************************************
```

```
PATH=/usr/bin:/usr/sbin:/usr/ucb:/bin
HOSTNAME=`uname -n`
SCRIPTS=/opt/admin/scripts
HWCONF=$SCRIPTS/hrdwspecs.sh
IWS_DIR1=/app/iplanet/iws
IWS_DIR2=/app/iplanet/ws4/enterprise
JRUN_DIR=/app/jrun
LDAP_DIR=/app/iplanet/ids
SM_DIR=/app/siteminder
ORA_TAB=/var/opt/oracle/oratab
SYB_DIR1=/app/sybase
SYB_DIR2=/syb/app
LEGATO_EXEC=/usr/sbin/nsrexecd
BACKUP_SRV=nypbck01

# List mounted file systems
function fslist
{
      mount -p | awk '
            $4 == "ufs" { print $3; }
            $4 == "vxfs" { print $3; }
      '
}

FSLIST=`fslist`

function system_status
{
print "\nStatus Taken at: "`date`
print "\n\n"
echo "$HOSTNAME up for: "`uptime | awk '{ print $3 , $4 }`
print "\n\n"
echo 'File System Size: \n'
df -k
print “\n”
if [ -f $LEGATO_EXEC ] ; then
print "\n\nChecking Backups:\n"
for i in $FSLIST
do
print "$i:"
mminfo -s $BACKUP_SRV -c $HOSTNAME -r 'savetime,volume,level' -q name=$i -t'1 week ago' -
ot
done
print “\n”
fi

if [ -d $IWS_DIR1 -o -d $IWS_DIR2 ] ; then
print "\nAre the IWS Listeners up?: \n"
ps -ef | grep [h]ttp
print “\n”
fi

if [ -d $JRUN_DIR ] ; then
print "\nAre the Jrun Instances up?: \n"
ps -ef | grep [-]start
print “\n”
fi
```

```
if [ -d $LDAP_DIR ] ; then
print "\nIs LDAP up?: \n"
ps -ef | grep [n]s-slapd
print “\n”
fi

if [ -d $SM_DIR ] ; then
print "\nIs SiteMinder up?: \n"
ps -ef | grep [s]mservauth
print “\n”
fi

if [ -f $ORA_TAB ] ; then
print "\nAre the Oracle Databases up?: \n"
ps -ef | grep [o]ra_
print "\nAre the Oracle Listeners up?: \n"
ps -ef | grep -i [l]istener
ps -ef | grep [o]rasrv
print “\n”
fi

if [ -d $SYB_DIR1 -o -d $SYB_DIR2 ] ; then
print "\nAre the Sybase Databases up?: \n"
ps -ef | grep [d]ataserver
ps -ef | grep [b]ackupserver
print “\n”
fi

#print "\nChecking Print Queues: \n"
#lpstat -o
#print "\nChecking Printer Status: \n"
#lpstat -t

print '\nWho has Switched Users?:\n\n'
tail -30 /var/adm/sulog
print '\n\nWho Last Logged into the System?:\n\n'
last | head -20
print '\n\nWho is Currently Logged on?:\n\n'
who -a | head –20

print '\n\nNetwork Status:\n'
print "netstat -i: \n"
netstat -i
print "\nifconfig -a: \n"
ifconfig -a
print '\nnetstat -rn:'
netstat –rn

print '\n\nChecking Mail Queue:\n\n'
mailq
print "\n"

# Call the hardware specifications script
if [ -x $HWCONF ]; then
      $HWCONF
fi
```

```
print '\nProcesses Currently Running (ps –ef):\n\n'
ps -ef
print '\n\nSYSTEM STATUS COMPLETE\n\n'
# End system_status function
}

if [ -z "$1" ]; then

system_status

else

           mail $1 <<EOF
From: $0
To: $1
Subject: System Status for $HOSTNAME
`system_status`

EOF

fi
exit 0
```

chmod 700 /opt/admin/scripts/status.sh

Hardware Audit Script (hrdwspecs.sh)

```
# vi /opt/admin/scripts/hrdwspecs.sh
#!/bin/ksh
#
# Solaris 2.X Hardware Specifications Script
# Purpose: Creates statistics for disk space, CPU, and
# memory
# Usage: Called by status.sh
# Dependencies: None
# Outputs: Standard out
#
#************************************************************

PATH=/usr/bin:/usr/sbin
DATE=`date '+%m-%d-%y%n'`

SVRNM=`uname -n`

# Ensure that temp files get cleaned up upon exit
trap '/bin/rm -fr $tmp; exit' 0 1 2 3 15
WRKFILE=/tmp/prog$$

df -k >> $WRKFILE

# Delete the first line and swap entry
```

```
{
vi $WRKFILE <<EOF
:1
dd
/swap
dd
:wq!
EOF
} > /dev/null

# If the cdrom drive is mounted, delete its entry too

CDR=`cat $WRKFILE | grep -c cdrom`

if [ "$CDR" -gt "0" ]; then

{
vi $WRKFILE <<EOF
/cdrom
dd
:wq!
EOF
} > /dev/null

fi

integer KTOTL=0
integer KUSED=0
integer KAVAIL=0

while read -r FS TOTL USED AVAIL CAP MNT
do

if [ "$TOTL" -gt "0" ]; then
((KTOTL = KTOTL + TOTL))
fi

if [ "$USED" -gt "0" ]; then
((KUSED = KUSED + USED))
fi

if [ "$AVAIL" -gt "0" ]; then
((KAVAIL = KAVAIL + AVAIL))
fi

done < $WRKFILE

# Translate KB to GB
((GTOTL = KTOTL / 1048576))
((GUSED = KUSED / 1048576))
((GAVAIL = GTOTL - GUSED))

echo " "
echo "$SVRNM Total Disk Space Usage:"
echo " "
echo "GB                USED            AVAIL"
echo "----------------------------------------"
```

```
echo "$GTOTL            $GUSED          $GAVAIL"
echo " "
echo " "
echo "$SVRNM CPU Specifications:"
echo " "
/usr/platform/`arch -k`/sbin/prtdiag | grep Configuration | awk {'print
$9,$10,$11,$12'}
echo " "
echo " "
echo "$SVRNM Memory Specifications:"
echo " "
/usr/platform/`arch -k`/sbin/prtdiag | grep 'Memory size' | awk {'print
$3,$4'}
echo " "
echo " "

rm $WRKFILE
exit 0
```

chmod 700 /opt/admin/scripts/hrdwspecs.sh

Performance Logging Script (perf_log.sh)

Purpose: Writes performance commands to log files. Uses vmstat, iostat, netstat, and other performance commands.
Usage: Run from crontab every 5 minutes, Monday through Friday, 9am - 5pm.
Warnings: Deletes any files in its output directory older than 14 days. If the size of the output directory exceeds 50 MB, it deletes files older than 7 days and sends an e-mail. Ensure that /var is mounted on a separate file system before using this script.
Dependencies: /etc/aliases – status (e-mail addresses of administrators)
Outputs: /var/adm/log/perf_log/*files*

vi /opt/admin/scripts/perf_log.sh

```
#!/bin/ksh
#
# Solaris 2.X Performance Log Script
# Purpose: Executes performance commands and saves
# the results to files named by the date.
# Usage: Execute the script from crontab at 5 minute intervals.
# Dependencies: None
# Outputs: Logfiles and e-mail
#
# Crontab Example (Mon - Fri 7am - 6pm):
# 01,06,11,16,17,21,26,31,36,41,46,51,56 9-17 * * 1-5 perf_log.sh
#
# WARNING: This script deletes any files older than 14 days in the
# $OUTPUTDIR directory!!!
#
# WARNING: Once the size of $OUTPUTDIR reaches above $MAXSIZ, any
# files older than 7 days are deleted in $OUTPUTDIR.
#
#****************************************************************

PATH=/usr/sbin:/usr/bin:.
DATESUFFIX=`date '+%m-%d-%y%n'`
# Where the log files will be written to
```

```
OUTPUTDIR=/var/adm/log/perf_log
# The directory this script resides in
ADMINDIR=/opt/admin/scripts
# The maximum size $OUTPUTDIR is allowed to reach before log files
# are deleted. (51200=50 MB)
MAXSIZ=51200

if [ !  -d $OUTPUTDIR ] ; then

      mkdir -p $OUTPUTDIR
fi

# The next variable can be set for multiple addresses
# (i.e. jsmith@yahoo.com,jsmith@hotmail.com)
MAILADD=status

# vmstat
date >> $OUTPUTDIR/vmstat.$DATESUFFIX
# The first line of vmstat is since reboot
vmstat 1 2 | tail -1 >> $OUTPUTDIR/vmstat.$DATESUFFIX

# iostat
date >> $OUTPUTDIR/iostat.$DATESUFFIX
iostat -xtnc >> $OUTPUTDIR/iostat.$DATESUFFIX

# netstat
date >> $OUTPUTDIR/netstat.$DATESUFFIX
netstat -i >> $OUTPUTDIR/netstat.$DATESUFFIX

# Process Data
date >> $OUTPUTDIR/ps.$DATESUFFIX
/usr/bin/ps -el -o pcpu,pmem,fname,rss,vsz,pid,stime >> $OUTPUTDIR/ps.$DATESUFFIX
date >> $OUTPUTDIR/ucbps.$DATESUFFIX
/usr/ucb/ps -aux >> $OUTPUTDIR/ucbps.$DATESUFFIX

# Kernel Data
date >> $OUTPUTDIR/kmastat.$DATESUFFIX
echo kmastat | crash >> $OUTPUTDIR/kmastat.$DATESUFFIX
date >> $OUTPUTDIR/kernelmap.$DATESUFFIX
echo "map kernelmap" | crash >> $OUTPUTDIR/kernelmap.$DATESUFFIX

# CPU Data
date >> $OUTPUTDIR/mpstat.$DATESUFFIX
mpstat >> $OUTPUTDIR/mpstat.$DATESUFFIX

# swap
date >> $OUTPUTDIR/swap.$DATESUFFIX
swap -l >> $OUTPUTDIR/swap.$DATESUFFIX 2>/dev/null

# /tmp (Running out of swap space)
# date >> $OUTPUTDIR/tmp_du.$DATESUFFIX
# du -sk /tmp >> $OUTPUTDIR/tmp_du.$DATESUFFIX
# date >> $OUTPUTDIR/tmp_ls.$DATESUFFIX
# ls -lt /tmp >> $OUTPUTDIR/tmp_ls.$DATESUFFIX

# Compress log files
```

```
for i in `find $OUTPUTDIR -mtime +1 -exec ls {} \; | grep -v .Z`
do

compress $i

done

# Delete any performance log files older than 14 days
find $OUTPUTDIR -mtime +14 -exec rm {} \;

# Ensure that log files do not take up more than 50MB
LOGDU=`du -sk $OUTPUTDIR | awk '{ print $1 }`

if [ "$LOGDU" -gt "$MAXSIZ" ]; then
           find $OUTPUTDIR -mtime +7 -exec rm {} \;
           mail $MAILADD <<EOF
From: $0
To: $MAILADD
Subject: Performance Log Size on `uname -n`
$OUTPUTDIR was $LOGDU KB. $0 does not
allow more than 50 MB of log files in this directory.
Log files older than 7 days have been deleted.
The current size of $OUTPUTDIR is `du -sk $OUTPUTDIR | awk '{ print $1 }` KB.
Thank you.
EOF
fi

exit 0
```

chmod 700 /opt/admin/scripts/perf_log.sh

Log Centralization Script (web_pull.sh)

```
# vi /opt/admin/scripts/web_pull.sh
#!/bin/ksh
#
# Solaris Web Log Pull Script
# Purpose: Downloads web server log files
# with FTP and SCP. Files older than 1 day
# are rotated and compressed. Sends e-mail
# if there is a failure.
# Usage: Execute from crontab (daily)
# Dependencies: None
# Outputs: Log files and e-mail
#***********************************************

PATH=/usr/sbin:/usr/bin:/usr/local/bin

# Webtrends directory
# (The log files are named by server)
LOGDIR=/weblogs/site1

# Archive directory
ARCHDIR=/webarch/site1
```

```
DATE=`date '+%m-%d-%y%n'`

UMASK=033

HOSTNAME=`uname -n`

# The next variable can be set for multiple addresses
# (i.e. jsmith@yahoo.com,jsmith@hotmail.com)
MAILADD=status

#
## Process the site1 logs...
#

# Move existing log files to the archive directory

for i in `/usr/bin/ls $LOGDIR`
do

gzip $LOGDIR/$i
mv $LOGDIR/$i.gz $ARCHDIR/$i.$DATE.gz

done

# Download today's log files

scp "admin@logsrv1.domain.com#22:/weblog/access.logsrv1" $LOGDIR
if [ $? -gt 0 ]; then
           mail $MAILADD <<EOF
From: $0
Subject: Web Server Log Centralization
The download of log files from logsrv1 to sunsrv01 has failed.
The files must be downloaded immediately. See $0 for details.
Once the files have been downloaded, click "Analyze Now" for each
site1 Webtrends profile. Otherwise, there will be a missing
day in the web statistics.
EOF
fi;

#
## Process the site2 logs...
#

# Webtrends directories
# (Log files are not named by server)
LOGDIR1=/weblogs/site2/websrv1
LOGDIR2=/weblogs/site2/websrv2

# Archive directories
ARCHDIR1=/webarch/site2/websrv1
ARCHDIR2=/webarch/site2/websrv2

MONTH=`date '+%b'`
DAY=`date '+%d'`
DAYMONTH=$DAY$MONTH
```

```
# Move existing log files to the archive directories

for i in `/usr/bin/ls $LOGDIR1`
do

gzip $LOGDIR1/$i
mv $LOGDIR1/$i.gz $ARCHDIR1/$i.$DATE.gz

done

for i in `/usr/bin/ls $LOGDIR2`
do

gzip $LOGDIR2/$i
mv $LOGDIR2/$i.gz $ARCHDIR2/$i.$DATE.gz

done

# Download today's log files

ftp -n logsrv2 <<EOF
       u sysact passwd
       prompt
       lcd $LOGDIR1
       cd /websrv1/iplanet/site2/SSL
       mget access.$DAYMONTH*
       lcd $LOGDIR2
       cd /websrv2/iplanet/site2/SSL
       mget access.$DAYMONTH*
       bye
EOF

# Check to see if the transfer completed

for i in $LOGDIR1 $LOGDIR2
do

if [ `ls $i | wc -l` -lt 1 ]; then
          mail $MAILADD <<EOF
From: $0
Subject: Web Server Log Centralization
The download of log files from logsrv2 to sunsrv01 has failed.
The files must be downloaded immediately. See $0 for details.
Once the files have been downloaded, click "Analyze Now" for each
site2 Webtrends profile. Otherwise, there will be a missing
day in the web statistics.
EOF
fi;

done

# Uncompress the log files so Webtrends can process them
gunzip $LOGDIR1/access*
gunzip $LOGDIR2/access*

# Ensure that the log files do not take up more than 100 MB.
```

```
# The maximum size $ARCHDIR is allowed to reach before notification
# is sent. (102400=100 MB)
MAXSIZ=102400

ARCHDIR=/webarch

LOGDU=`du -sk $ARCHDIR | awk '{ print $1 }`

        if [ "$LOGDU" -gt "$MAXSIZ" ]; then
           mail $MAILADD <<EOF
From: $0
Subject: Web Log Size on $HOSTNAME
$ARCHDIR is $LOGDU KB. $0 notifies of
more than 100 MB of log files in this directory.
Thank you.
EOF
        fi

exit 0
```

chmod 700 /opt/admin/scripts/web_log.sh

Volume Manager Configuration Script (vmconfig.sh)

Purpose: Saves Volume Manager configuration. Logs “vxdisk list”, “vxprint –ht”, “df –k”, and /etc/vfstab.
Usage: Run from crontab every night.
Dependencies: None
Outputs: /var/adm/log/backup/vmsa.log
/var/adm/log/backup/*diskgroup*.conf

vi /opt/admin/scripts/vmconfig.sh

```
#!/bin/sh
#
# Volume Manager Configuration Script
# Solaris 2.X
#
#*********************************************

# Where the log files will be written to
OUTPUTDIR=/var/adm/log/vmsa

LOGFILE=$OUTPUTDIR/vmsa.log
cp /dev/null $LOGFILE

if [ !  -d $OUTPUTDIR ] ; then
      mkdir -p $OUTPUTDIR
fi

{
echo "Volume Manager Configuration for `uname -n` on `date`"
echo " "
echo "This is the output of vxdisk list:"
echo " "
vxdisk list
echo " "
echo "This is the output of vxprint -ht:"
```

```
echo " "
vxprint -ht
echo " "
echo "This is the output of df -k:"
echo " "
df -k
echo " "
echo "This is the contents of /etc/vfstab:"
echo " "
cat /etc/vfstab
echo " "
echo " "
} >> $LOGFILE

# Backup the Volume Manager configurations by disk group
vxprint -g rootdg -vpshm > /var/adm/log/backup/rootdg.conf

# NOTE: To restore the disk group configuration:
# "vxmake -g <disk_group> -d filename"

exit 0
```

chmod 700 /opt/admin/scripts/vmconfig.sh

Install Security Audit Script (sec_audit.sh)

Download and install the CIS Solaris Benchmark Tool from http://www.cisecurity.org.

pkgadd -d CISscan

vi /opt/admin/scripts/sec_audit.sh

```
#!/bin/ksh
#
# Solaris 2.X Security Audit Script
# Purpose: Perform a security audit of the server each month.
# Dependencies: /opt/CIS/cis-scan
#               CIS Solaris Benchmark Tool
#               http://www.cisecurity.org
# Useage: Execute from command line or crontab each month.
# Outputs: Logfiles and e-mail to the operations team
# Crontab example:
# 01 7 2 * * /opt/admin/scripts/sec_audit.sh jsmith@yahoo.com
#
# WARNING: THIS SCRIPT DELETES ANY FILES OLDER
# THAN 35 DAYS in the $LOGDIR directory!!!
#
#*****************************************************************

PATH=/usr/sbin:/usr/bin:/usr/local/bin

# The next variable can be set for multiple addresses
# (i.e. jsmith@yahoo.com,jsmith@hotmail.com)
MAILADD=status
```

```
HOSTNAME=`uname -n`

LOGDIR=/var/adm/log/cis-tool

# Make sure $LOGDIR exists

if [ ! -d $LOGDIR ] ; then
      mkdir -p $LOGDIR
      chmod 700 $LOGDIR
fi

DAY=`date '+%d'`
MONTH=`date '+%m'`
YEAR=`date '+%Y'`
DATE=$YEAR$MONTH$DAY

# Execute the CIS toolkit

/opt/CIS/cis-scan > /dev/null

# Move the log files

mv /opt/CIS/cis-ruler-log* $LOGDIR

# Delete any performance log files older than 35 days
find $LOGDIR -mtime +35 -exec rm {} \;

function security_audit
{
print "\nAudit Taken at: "`date`
print '\n\nWho has Switched Users?:\n\n'
tail -100 /var/adm/sulog
print '\n\nWho Last Logged into the System?:\n\n'
last | head -100
print '\n\nWho is Currently Logged on?:\n\n'
who -a | head -20
print "\n\nThis section contains the findings of a vulnerability assessment conducted"
print "by the CIS Solaris Benchmark and Scoring/Scanning Tool
(http://www.cisecurity.org).\n"
egrep "^Negative" $LOGDIR/cis-ruler-log.$DATE-*
print '\n\nProcesses Currently Running:\n\n'
ps -ef
print '\n\nSECURITY AUDIT COMPLETE\n\n'
# End security_audit function
}

# Send the results

if [ -z "$1" ]; then

security_audit

else

           mail $1 <<EOF
From: $0
To: $1
```

```
Subject: $HOSTNAME Security Audit

`security_audit`

EOF

fi

# Ensure that the log files do not take up more than 50 MB

# The maximum size $OUTPUTDIR is allowed to reach before log files
# are deleted. (51200=50MB)
MAXSIZ=51200

LOGDU=`du -sk $LOGDIR | awk '{ print $1 }`

        if [ "$LOGDU" -gt "$MAXSIZ" ]; then
           mail $MAILADD <<EOF
From: $0
Subject: Web Log Size on $HOSTNAME
$LOGDIR is $LOGDU KB. $0 notifies of
more than 50 MB of log files in this directory.
Thank you.
EOF
        fi
exit 0
```

chmod 700 /opt/admin/scripts/sec_audit.sh
/opt/admin/scripts/sec_audit.sh | more (to test)

Added the Monitoring/Logging Scripts to Crontab

```
# crontab –e
# The Sendmail daemon is not running - This tells it to send mail out
05,20,35,50 * * * * /usr/lib/sendmail –q
# Monitoring scripts
22,52 * * * * /opt/admin/scripts/mon_fs.sh > /dev/null
12,42 7-18 * * * /opt/admin/scripts/mon_procs.sh > /dev/null
13,43 7-18 * * 1-5 /opt/admin/scripts/mon_prf.sh > /dev/null
17,47 * * * * /opt/admin/scripts/mon_srv.sh > /dev/null
05 21 * * * /opt/admin/scripts/maildu.sh
14,29,44,59 * * * * /opt/admin/scripts/rtlgn.sh > /dev/null
01 8 * * 1-5 /usr/local/etc/logcheck.sh > /dev/null
# Reporting scripts
36 7 * * 1-5 /opt/admin/scripts/status.sh status > /dev/null
01 7 2 * * /opt/admin/scripts/sec_audit.sh status
# Logging scripts
0 0 * * * /etc/security/newauditlog.sh
0 21 * * * /opt/admin/scripts/vmconfig.sh 1>/dev/null 2>/dev/null
01,06,11,16,17,21,26,31,36,41,46,51,56 9-17 * * 1-5 /opt/admin/scripts/perf_log.sh
```

Created Application Start Scripts

In /opt/admin/scripts, created fullup.sh and fulldown.sh to stop and start all applications on the server. Once fulldown.sh has been run, all that remains is to halt or reboot the server. This ensures that software is shutdown in the proper order, with the proper time dependencies. Linked the scripts from /etc/rc3.d and /etc/rc0.d.

REBOOTED THE SYSTEM

A reboot is required for the settings to take effect.

```
# /usr/sbin/shutdown –i6 –g0 -y
```

BACKED UP THE SYSTEM

Install the backup agent and restore a few files from backup.

ADDITIONAL CONSIDERATIONS

Solaris Hardening Tools

Solaris Hardening Toolkit
http://wwws.sun.com/software/security/jass

Titan Security Scripts
http://www.fish.com/titan

YASSIP Security Scripts
http://www.yassp.org

Fix Modes

Fix modes hardens the default permissions of Solaris. It should be rerun after patching or application install.

http://wwws.sun.com/blueprints/tools/FixModes_license.html

TCP Wrappers

TCP Wrappers restricts connection to inetd services by IP address. It also logs all access attempts.

http://www.sunfreeware.com
http://rr.sans.org/unix/TCP_wrappers2.php

Tripwire

Tripwire is a file integrity checker, used for intrusion detection. Consider using it to detect alteration of binaries, configuration files and web content. I recommend the commercial version.

Commercial Tripwire: http://www.tripwire.com
Tripwire Academic Source Release (ASR): http://www.tripwire.com//products/tripwire_asr
CERT article: http://www.cert.org/security-improvement/implementations/i002.02.html

chkrootkit

Quoting from the site, "chkrootkit is a tool to locally check for signs of a rootkit." Like a virus checker, it only knows about the root kits that were available when it was produced. Run chkrootkit daily from cron.

I recommend renaming the chkrootkit script to something less obvious (i.e. appush.sh). If an intruder gains root access, they may see the cron entry and disable the script from inside.

http://www.chkrootkit.org
http://rr.sans.org/malicious/chkrootkit.php

Solaris Role-Based Access Control (RBAC)

Solaris RBAC enables system administrators to pass administrative access to users. It also logs access.

http://wwws.sun.com/software/whitepapers/wp-rbac
http://www.samag.com/documents/s=7667/sam0213c/0213c.htm

Solaris IP Multipathing

For high availability, consider setting up Solaris IP multipathing, using separate switches. Multipathing ensures that an IP stays up in the event of a NIC failure. At a minimum, this requires 2 NICs with 2 IP addresses for both (for a total of 4). Essentially, a failure causes the defective interface to be unplumbed from the first NIC and plumbed to the second. As an added bonus, outbound load balancing is also included.

Solaris 8 2/02 IP Network Multipathing Administration Guide
http://docs.sun.com/?q=IP+Network+Multipathing+&p=/doc/816-0850

Remote System Control Cards

For remote console access and notifications of hardware failures, set up the RSC card. Administrators can access the server's console remotely through telnet or the RSC client GUI. The console can also be accessed locally through the RSC serial port. RSC stays up when the server is powered off. RSC cards only ship with the latest Sun servers (i.e. Sun 250, 280, 480, 880, etc). Consider using a switched hub if many RSC cards are in use. They are rarely used and the expense of numerous ports on the network switch can add up quickly. NOTE: RSC cards run at 10 Mbps half duplex. Ensure that the switch does not force 100 Mbps full duplex.

Sun Remote System Control (RSC) Installation Guide
http://docs.sun.com/?q=rsc&p=/doc/816-3886-10

Sun Remote System Control (RSC) 2.2 User's Guide

http://docs.sun.com/?q=rsc&p=/doc/816-3314-10

Sun Fire V480 Server Administration Guide
http://docs.sun.com/?q=v480&p=/doc/816-0904-10

Solaris Fingerprint Database

The Solaris Fingerprint Database contains MD5 encrypted signatures of original Solaris files. It compares the system's binaries, patches and "unbundled products" to the database and notifies of discrepancies.

http://www.sun.com/solutions/blueprints/0501/Fingerprint.pdf

The Coroners Toolkit

From the site, "TCT is a collection of programs by Dan Farmer and Wietse Venema for a post-mortem analysis of a UNIX system after break-in".

http://www.porcupine.org/forensics/tct.html
http://rr.sans.org/incident/TCT.php

Harden Applications

Don't forget to harden the system's applications. Some basic steps include: disable unnecessary services, remove default applications, change default user ids and passwords, harden permissions, and configure logging. Search the Internet for specific hardening configurations for each application.

Patching

All systems should be patched at least every 6 months. Start by patching in development, then the remaining physical environments (i.e. UAT and production). Sun provides Patch Check to produce a report of currently installed patches versus those that are available (http://sunsolve.sun.com/pub-cgi/show.pl?target=patchk). A current patchdiag.xref file should be downloaded each time the tool is used. It contains the list of current patches. Patch Check will eventually be replaced by Patch Manager (http://www.sun.com/service/support/sw_only/patchmanagement.html).

Monitoring

Remotely monitor the system for availability (at a minimum). Additional considerations include host and network monitoring, vulnerability assessment, intrusion detection, remote url and application monitoring (i.e. application login, connection pools, databases, etc).

System Operations Guide

Create an OPS guide to provide continuity for the system. At a minimum, it should detail: (a) stop and start the system and its applications, (b) application administration, (c) replace a failed hard drive, and (d) restore the system from backup.

REFERENCES

Armoring Solaris
http://www.enteract.com/~lspitz/armoring.html

Securing Solaris Servers - A Checklist Approach
http://www.usenix.org/sage/sysadmins/solaris/index.html

Securing Solaris
http://www.securityfocus.com/focus/sun/articles/securing.html

Harden Solaris
http://www.boran.com/security/sp/Solaris_hardening.html

Solaris Security Guide
http://www.sabernet.net/papers/Solaris.html

Sun's Solaris Security FAQ
http://www.itworld.com/Comp/2377/security-faq

Fix Solaris
http://fixsolaris.sunhelp.org/fixsolaris.txt

Solaris Operating Environment Security
http://www.sun.com/blueprints/0100/security.pdf

Solaris 8 System Administrator Collection
http://docs.sun.com/db/coll/47.11

Sun Product Documentation
http://docs.sun.com

Sunsolve Online
http://sunsolve.sun.com/pub-cgi/show.pl?target=home

Last updated on: November 7, 2002