Global Information Assurance Certification Paper

Securing a Multi-User Solaris 8 SPARC System

Software/Hardware Preparation

Scenario

This paper describes the process for securing a Sparc based server with Solaris 8 operating system for remote access. The users often travel and work from home, and they needs to have access to this machine over the Internet for transfering files and connecting to other machines in the internal network computing resources. The users should be able to connect to this machine, store and transfer files, run their own applications in the home directories, and launch X-based applications from other X-terminals (the system is an X client).

Security will be implemented considering that the system will be a general multi-user system. The focus of this paper is on protecting sensitive information that may travel over the internet using OpenSSH and TCP Wrappers. In addition, measures are taken to *protect* the system by minimizing the potential holes and protecting the file system to prevent unauthorized access, and configuring the system to prevent and minimize the damages from denial of service attacks by internal and external threats while maintaining the availability of necessary services, manageability and system performance.

Hardware Preparation:

Two Sparc-based machine is required to carry out the steps presented in this paper:

- **One Sparc based machine to be used as a UNIX system to be used as an access point over ssh.**
- **One Sparc based machine to be used as a development platform to download, verify and compile necessary software components to be installed in the above system with an Internet connection (http and ftp for viewing the manuals in the web and to download necessary software) .**

Both machines are UltraSparc 10 with 256Mb RAM, and 10 Gb internal SCSI hard disk. A CD-Rom drive and an 8mm tape drive are used to install Solaris 8 operating system and transfer files from the development machine to the production machine.

Software Preparation

Solaris 8 Operating System CDs and Solaris 8 Operating System Update 07/01 are needed carry out the steps presented in the following sections. Follow the steps for each hardware platforms: development and production ssh server.

Development System

1. Install Solaris 8 operating system with the Development Cluster option.

2. In addition the operating system CDs (disk 1 and 2) required to install the operating system, the following software should be obtained to follow the steps presented in this paper; download, or prepare the following software in the specified format (in a suitable directory). If the link does not work, go to http://www.sunfreeware.com and other web sites to locate the appropriate links updated versions of these software:

   GCC Compiler

   Precompiled binaries can be obtained and installed to compile TCP Wrappers 7.6 and OpenSSH 3.0p1. The Solaris Software Companion CD included with Solaris 8 operating system Update 07/01 provides a convenient mechanism to install the compilers and other necessary software components such as flex, bison, binutils, gmake and other library files.

   TCP Wrappers 7.6

   Purpose: This software is a wrapper program used to monitor and control the access to TCP based services such as telnet, ftp, etc. Even though it can be used to improve the security posture significantly, it is still vulnerable to IP spoofing attacks. To improve the security of remote access to these services, this paper recommends that it is used to manage OpenSSH connections. See next section for the description of OpenSSH.

   Source download: ftp://ftp.porcupine.org/pub/security/index.html
   Source file name: tcp_wrappers_7.6.tar.gz

   OpenSSH 3.0p1

   This software is used to replace insecure network services such as rlogin, rcp, ftp, etc. These services pose vulnerability because they transmit sensitive acess information unencrypted. OpenSSH provides an asymmetric encription based authentication mechanism to authenticate the hosts. In addition, proper router and firewall configuration, and the use of Intrusion Detection Systems (IDS) can further improve the security.

   Source download: http://www.openssh.com/portable.html
   Solaris precompiled binary download (v2.9): http://www.sunfreeware.com
   Source file name: openssh-3.0p1.tar.gz

   FixModes

   Purpose: A solaris permissions hardening scripts. It removes group and world writable permissions. The changes are made to all files and directories that are

listed in in /var/sadm/install/contents with the exception of those listed in exceptions.h file.
Source and Precompiled download
http://www.sun.com/blueprints/tools/FixModes.html
Source file name: FixedModes.tar.Z

Solaris 8 Security Patches

Install recommended and security patches from SunSolve Online (http://sunsolve.sun.com).

File name: 8_Recommended.zip

3. Install GCC compiler. Insert the CD onto the development system, and double-click "installer", follow the graphical user interface to install the following group of applications.

- Development/Languages
- Development/Libraries
- Development/Tools

The gcc compiler binary is located in ***/opt/sfw/bin*** by default.

4. Set the PATH environment variable.

```
# PATH = $PATH:/opt/sfw/bin:/usr/ccs/bin
# export PATH
```

Now, the development machine is ready for compiling the TCP wrappers, OpenSSH and Fixmodes.

Solaris OS Installation (Target Production Host)

Installation Precaution

First, install the operating system on the production server without connecting to the network, as the production system could possibly be compromised during the installation.

Minimization

Added software components often translate into more labor and complexity to secure and maintain the software. Minimizing the software components down to bare necessities is not an easy task either because most software application vendors do not precisely state which packages are required to run their application. For the initial installation procedure in this paper, "Solaris Operating System Minimization for Security", November 2000 from Sub BluePrints Online was used to identify the necessary packages that needs to be installed and many of the unnecessary packages that can be removed.

Starting at the OK> prompt, reboot the system using the cd-rom.

```
OK> boot cdrom
```

1. Select a language: 0
2. Network: yes
3. Use DHCP: No – It will be a Internet ready server.
4. Hostname: <enter hostname>
5. IP Address: <enter ip>
6. Part of Subnet: Yes
7. Subnet mask: <enter subnet>
8. IP v6: <yes if IP v6 support is required>
9. Configure Kerberos Security: No
10. Name Service: none
11. Time zone: Geographic Region
12. Select appropriate region
13. Select Date and Time
14. Select "Initial Installation"
15. Select Geographic Regions, and select only the languages that are necessary.
16. "U.S.A. (en_US_ISO8859-1)"
17. Select "Core System Support"
18. Require 64-bit support? yes
19. If yes, click select to include Solaris 64bit support.

Assign space for each of the following partitions:

- The minimum required operating system components will take less than 50 mb of the root partition, but the extra space provides protection against root partition being filling up, and crashing the system. Some may say it's too much.

/ 512 Gb

- Assign enough space to holds log files.

/var 512 Mb

- Assign space for user home directories.

/export 6 Gb

- Assign space for installing applications

/usr 1.5 Gb
/opt 1 Gb

- Assign swap space double the size of RAM

swap 512 Mb

Add other packages such as NTP and zlib.

Using this command,
pkgadd -d /cdrom/Solaris_8/Product <Package Name>

Install the following additional packages:

TermInfo Database and System Accounting Package

SUNWter
SUNWaccr
SUNWaccu
SUNWzlib
SUNWwice

NTP

SUNWntpu
SUNWntpr

For OpenSSH X Tunneling

SUNWxcu4
SUNWxcu4x
SUNWxwplt
SUNWxwplx
SUNWxwrtl
SUNWxwrtx
SUNWswmt
SUNWxwice
SUNWxwicx

Then, install 109667-03 patch that fixes several serious NTP bugs, and configure at least three time servers by editing /etc/inet/ntp.conf:

```
server <time server ip 1> prefer
server <time server ip 2>
server <time server ip 3>
```

"prefer" argument should assigned to the time server that is believed to be most accurate, but the other two server can check against the preferred time server when it significantly deviates from the other two time servers.

If external ntp server is outside the firewall, NTP port 123 should be open.

Security Patches

Install recommended and security patches from SunSolve Online (http://sunsolve.Sun.COM/pub-cgi/show.pl). Once the patch cluster is downloaded from the site, verify the checksum of the downloaded files using a MD5 utility.

File name: 8_Recommended.zip (As of 12/11/2001)

Unzip 8_Recommended.zip file, and run *install_cluster* script.

It is important that security patches are applied before taking other system hardening measures because some patches may overwrite the changes that are made. In addition, the application of some patches will result in the installation of previously removed software components. The patches can be selectively applied, but it is easier at this stage to install the patch cluster, and then remove the unnecessary packages identified in the next section.

Removal of Unnecessary Packages

Appendix A contains the list of packages that can be removed from the system. The source of this list was taken from http://www.enteract.com/~lspitz/jump-remove.txt, and this list was compiled from JASS security toolkit from Sun Microsystems web site (http://www.sun.com/blueprints/tools). The list in Appendix A developed for Netscape iPlanet Webserver, but this list is used here because this example will require even less number of packages than iPlanet Webserver.

```
# /usr/sbin/pkgrm <package name>
```

When additional applications are installed later, identify all required packages for the Internet application to run as specified by the vendor. Remember to identify and apply the latest security patches after necessary Operating System packages are installed.

Configure Gateway

Create /etc/defaultrouter file that contains the IP address of the router.

```
# echo XXX.XXX.XXX.XXX > /etc/defaultrouter
```

Create /etc/notrouter to disable router capabilities.

```
# touch /etc/notrouter
```

Configure Name Service

Because the machine was off-line during the installation, name service was not configured. At this point resolve.conf should be created, and it should contain

```
domain <company>.com
nameserver <ip of the dns>
search <company>.com
```

/etc/nsswitch.conf should contain

```
hosts: files dns
```

so that /etc/hosts file has the priority just in case DNS is compromised.

Console Security

For the maximum security of a host, it should be placed where physical access to the system is restricted. At the minimum, it needs to be locked, and it should only be accessible by the administrator who is maintaining the host.

Also, the system should be protected against the power outage, and occasional surges. The server should be connected to a UPS (uninterruptable power supply) and appropriate surge protector.

The server should stays unavailable until the administrator restores the system. This measure may increase the downtime, but when the machine is shutdown, the administrator gets the first chance to troubleshoot the system.

eeprom security-mode=full
eeprom security-password=

Note: "Sun Operating System Security ", April 2001 states that EEPROM password can not be recovered, and SunService needs to be contacted for a new EEPROM. Some EEPROMs in Sparc machines can be reset by physically powering off, powering back on, and holding [STOP] [N] keys simultaneously until EEPROM prompts for a new password.

To monitor password login guessing add the following line to an initialization script.
eeprom security-badlogins=0

Later, typing
eeprom security-#badlogins
will reveal the number of bad logins.

In addition, disable the keyboard abort sequence by uncommenting the following line in /etc/default/kbd file

KEYBOARD_ABORT=disable

This measure will prevent users from using Stop-A sequence to gain access to boot prompt.

Login and Password

Edit /etc/default/login file to set the login policy.

```
# The root user can only log in directly from the
# system console. PermitRootLogin parameter in
# OpenSSH server configuration will actually limit the
# ability to log on as root over the network.
CONSOLE = /dev/console

# Set the initial PATH settings for root to ensure that
# PATH initially points to the right directories.
# The administrator will have to use su when they are using
# ssh to login over the network.
SUPATH = /usr/sbin:/usr/bin

# Set the initial shell file creation mode so that the user
# created files are not writable by group, and not accessible
# at all by other users
UMASK = 027

# log failed login attempts
SYSLOG_FAILED_LOGIN = 0
```

Make sure the umask is not overriden to a value less strict than 027 in the following files:

```
/etc/.login
/etc/profile
$HOME/.cshrc
$HOME/.login
$HOME/.profile
```

where $HOME is the home directory for each users (verify as users are added to the system).

Set the password policy by editing /etc/default/passwd file. Set the maximum time period for a passwd to 13 weeks so that user passwords will change every 3 months or so, and the minimum time period to 1 week to prevent users from changing password multiple times in a short period time to set the passord back to the original password. The password length makes the passwords more difficult to guess to prevent not only manual guesses, but also to make it harder for a hacker to crack password hashes in case he/she was able to obtain them.

```
# /etc/default/passwd file
MINWEEKS = 1
MAXWEEKS=13
PASSLENGTH=8
```

Boot Scripts

Disable the following boot script using the convention used by Hal Pomeranz's Solaris "Security Step-by-Step". Rename these files by attaching ".NO" in the beginning so that the new file name would be ".NO<previous filename>".

- Disable NFS because NFS traffic flows in cleartext.

```
/etc/rc2.d/S73nfs.client ->   /etc/rc2.d/.NOS73nfs.client
/etc/rc3.d/S15nfs.server ->   /etc/rc3.d/.NOS15nfs.server
/etc/rc2.d/K28nfs.server ->   /etc/rc2.d/.NOK28nfs.server
/etc/rc2.d/S74autofs ->  /etc/rc2.d/.NOS74autofs
```

- RPC (Remote Procedure Call) uses IP address and UID for authentication, and is not generally secure. This installation does not require RPC.

```
/etc/rc2.d/S71rpc        ->   /etc/rc2.d/.NOS71rpc
/etc/rc2.d/S76nscd       ->   /etc/rc2.d/.NOS76nscd
```

- This machine is not a e-mail server. E-mails can be sent out from this machine without a mail server daemon running.

```
/etc/rc2.d/S88sendmail   ->   /etc/rc2.d/.NOS88sendmail
```

- This machine is not configured to use ldap

```
/etc/rc2.d/S71ldap.client ->  /etc/rc2.d/.NOS71ldap.client
```

- Solaris auto-configuration services are not needed.

```
/etc/rc2.d/S30sysid.net  ->   /etc/rc2.d/.NOS30sysid.net
/etc/rc2.d/S71sysid.sys  ->   /etc/rc2.d/.NOS71sysid.sys
/etc/rc2.d/S72autoinstall ->  /etc/rc2.d/.NOS72autoinstall
```

Filesystem Mount Options

Security of the system can be enhanced by using read-only and nosuid mount option. No suid files should be present outside root (/) or /usr partition. In the target host, software application that require suid attributes in /opt partition should be avoided. /var directories and /export/home needs to have "write" access, but suid files should not be executed from those directories.

Edit /etc/vfstab so that the following directories and partitions will mount with the following options:

mount point	**mount option**
/	-
/usr	ro (read only)
/var	nosuid (suid does not work)
/export/home	nosuid (suid does not work)
/opt	nosuid, ro (read only)

Later, when additional software needs to be installed under /usr or /opt partition, temporarily change the mount option to rw, shutdown and restart the system. /usr directory can only be unmounted by shutting down the system.

Until reboot, additional software (OpenSSH) can still be installed because read-only has not taken effect. In addition, OpenSSH requires its configuration directory to be writable, and it will reside in /etc/ssh directory.

Legal Notifications

/etc/motd file will be displayed in the incoming ssh connections. This file needs to contain warning messages that the activities in the system will be monitored for unauthorized and inappropriate use.

Vendor Default Accounts

The following users except should be either locked or assigned "no password" so that they could not be used to log in. They also need to be assigned an unusable shell:

daemon
bin
sys
adm
lp
nobody
nobody4
uucp
nuucp
listen

 # passwd -l <username> will assign "*LK*" in the password field.
 # passwd -e <username>
when prompted to enter the new shell type,
`New shell:` **/bin/false**

Auditing

Enable the loggin of failed login attempts (via /usr/bin/login) by creating an empty loginlog file.

```
# touch /var/adm/login
# chmod 600 /var/adm/login
# chown root:sys /var/adm/loginlog
```

Repeat the above procedure for /var/log/authlog file

In addition, add the following line to /etc/syslog.conf. This step is required to enable the logging of unsuccesful login attempts.

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

Edit /etc/lib/newsyslog script to configure log rotation. See Appendix C for the updated newsyslog script. Appendix C rotates syslog, authlog, and loginlog logs, and sets secure permissions on those files that that they may not be exploted.

Consider configuring syslog.conf that a central loghost that will store the logs by adding the following line to the syslog.conf file. This provides a back-up in case the system is compromised, and the attacker has deleted the log information on the host.

```
auth.warning ifdef (`LOGHOST`, /var/log/authlog, @loghost)
```

Configure the loghost to accept the logs from this host.

Sendmail

Sendmail server was disabled at the next reboot because the sendmail server boot script has been renamed in the previous section. However, it needs to be able to send mail.

Download sendmail file from http://www.sun.com/blueprints/tools/ to the development box, and place it in /etc/default directory in the production system.

The content of this configuration file is included with this report in Appendix B. The installation of this configuration file allows the periodic flushing (15 min) of the mail queues. This is an undocumented approach that can be used instead of adding "sendmail –q" implemented in the cron job in Solaris 8[1]. `MODE=""` setting /etc/default/sendmail enables this.

Network Settings

The following lines should be added at the end of /etc/init.d/inetinit to protect the system from common TCP/IP based attacks on cache, spoofing, and other DoS (Denial of Service) attacks.

```
# Reduce the arp cache, and routing table entry timeout
# to 1 minute by running the following commands.
# In a congested network, increase the interval
#(specified in milliseconds) if a drastic
# performance degradation is observed.

ndd -set /dev/arp arp_cleanup_interval 60000
ndd -set /dev/ip ip_ire_arp_interval 60000

# disable responses to echo multicasts/broadcasts
```

[1] Alex Noordergraaf and Keith Watson, Sun BluePrints OnLine, April2001

```
# that can produce a large number of packets
ndd -set /dev/ip6 ip6_respond_to_echo_multicast 0
ndd -set /dev/ip ip_respond_to_echo_broadcast 0


# disable responses to timestamps (unicast and broadcast)
ndd -set /dev/ip ip_respond_to_timestamp_broadcast 0
ndd -set /dev/ip ip_respond_to_timestamp 0

# Verify address mask broadcast is turned off
ndd -set /dev/ip ip_respond_to_address_mask_broadcast 0

# Attacker might forge redirect errors to fool the
# system that a new router is in place.
ndd -set /dev/ip ip_ignore_redirects 1
ndd -set /dev/ip ip6_ignore_redirects 1

# router functions
ndd -set /dev/ip ip_send_redirects 0
ndd -set /dev/ip6 ip6_send_redirects 0
ndd -set /dev/ip ip_forwarding 0
ndd -set /dev/ip6 ip6_forwarding 0
ndd -set /dev/ip hme0:ip_forwarding 0

# a measure against connection exhaust attack.
# indirectly shorten the connection timeout setting
# much more resource is required (default = 128).

ndd -set /dev/tcp tcp_conn_req_max_q 1024

# a measure agains SYN flood attack. Default is
# 1024.
ndd -set /dev/tcp tcp_conn_req_max_q0 4096
```

Edit /etc/default/inetinit to that improved TCP algorithm is utilized to prevent TCP sequence prediction.

```
TCP_STRONG_TSS = 2
```

and, add the folllowing to the end of /etc/init.d/inetinit to enable the above change.

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

Other Miscellaneous Configuration Files

.rhosts and hosts.equiv files

Lock down the .rhosts and hosts.equiv files and hosts.equiv files

```
# echo "-" > /.rhosts
# echo "-" > /etc/hosts.equiv
# chmod 600 /.rhosts
# chmod 600 /etc/hosts.equiv
# chown root:sys /.rhosts
# chown root:sys /etc/hosts.equiv
```

/etc/pam.conf

In addition, remove those rlogin or other r-command (rsh and rlogin) related references from pam.conf file such as the following lines.

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

/etc/system

Set noexec_user_stack and noexec_user_stack_log to the following values to protect against the buffer overflow attack:

```
set noexec_user_stack = 1
```

- disables the users from executing code on the system stack.

```
set noexec_user_stack_log =1
```

- logs any attempts to execute code on the stack.

Set maxuprc parameter to 128 on this multi-user machine to limit the number of processes per non super-user to limit resource consumption.

```
set maxuprc=128
```

Core file can be used to recover passwords or other restricted information. The following setting essentially disables the core generation

```
set sys:coredumpsize=0
```

Install OpenSSH and TCP Wrappers

Further Preparation on the Development System

Make sure that "make" and "ar" tools are available in the PATH variable.

rename "/opt/sfw/bin/gmake" to "/opt/sfw/bin/make".

```
# mv /opt/sfw/bin/gmake /opt/sfw/bin/make
```

Set the PATH variable to include the path to "ar" and "make".

```
# PATH = $PATH:/usr/ccs/bin:/opt/sfw/bin
# echo $PATH

/usr/sbin:/usr/bin:/usr/ccs/bin:/opt/sfw/bin

# which ar
```

```
/usr/ccs/bin/ar
# which make
/opt/sfw/bin/
```

Set CC variable to "gcc".

```
# CC = /opt/sfw/bin/gcc
# export CC
```

TCP Wrappers

From the development system, unpack TCP Wappers gziped tar ball.

```
# gzip –d tcp_warppers_ipv6_7.6.tar.gz
# tar xf tcp_wrappers_ipv6_7.6.tar.gz
```

Uncomment the REAL_DAEMON_DIR path from the Makefile.

```
# SysV.4 Solaris 2.x OSF AIX
REAL_DAEMON_DIR=/usr/sbin
```

Modify the FACILITY variable so all logs will be handled by the authentication log facility

```
# The LOG_XXX names below are taken from the
/usr/include/syslog.h file.

# FACILITY= LOG_AUTH    # LOG_AUTH is better place to send log
```

Run
make sunos5

Copy the following files in the current directory to the specified location in the development machine:

- libwrap.a file to /usr/local/lib
- tcpd.h file to /usr/local/include

OpenSSH

To install OpenSSH on the development system, OpenSSL is required. The version used here is v0.96b.

ungzip and untar the source code distribution package.

run config

```
# sh config
…
# make
…
```

```
# make install
…
```

The software is installed under the following directories.

/usr/local/bin/	executable binary utilities
/usr/local/lib/ssl	static and shared crypto/ssl libraries
/usr/local/ssl	documentation and other tools.

Unpack openssh3.0p1.tar.gz file, and cd into openssh-3.0p1 directory.

```
# gzip –c openssh3.0p1 | tar -xf
# cd openssh-3.0p1
```

set CFLAGS environmental variable to –I/usr/local/include so that TCP Wrapper header (tcpd.h) files can be accessed, and set LDFLAGS variable to –L/usr/local/lib.

```
# CFLAGS=”-I/usr/local/ssl/include”
# LDFLAGS = “-L/usr/local/lib”
# sh configure –prefix=/usr/local
–sysconfdir=/etc/ssh
–with-tcp-wrappers
–without-rsh
–with-pam
```

- -prefix decides the top directory where OpenSSH directories are laid down.
- -sysconfdir decides where OpenSSH files are stored. It is stored in the root partition because /usr partition will be mounted “read-only”, and the configuration should be writable even though it has to be protected.
- -with-tcp-rappers includes libwrap.a (TCP Wrapper library) so that ssh can be configured using /etc/hosts.allow and /etc/hosts.deny).
- -without-rsh flag will not let the system use rsh when ssh fails.
- -with-pam enables PAM suport.

Refer to the following web-site for more available configuration options.
ftp://ftp.ca.openbsd.org/pub/OpenBSD/OpenSSH/portable/INSTALL

```
# make
…
```

The performance of ssh may improve when PRNGD 0.9.19 is installed. It is a random number generator pool that can generate random numbers required to generate keys, but the author is not certain about its security implications where and how these random numbers might be stored on the system. See http://www.mail-archive.com/openssl-announce@openssl.org/msg00024.html for details.

Transfer the following files in the current directory (where the source code resides) to the specified location in the target host:

- sshd to the production machine's /usr/local/sbin directory.
- sftp-server to the production machine's /usr/local/libexec directory.
- ssh_prng_cmds, ssh_config, sshd_config to /etc/ssh directory
  /etc/ssh/sshd_config is the sshd configuration file. See Appendix D for sshd_config file and ssh_config file. The comments in the Appendix briefly describe the purpose of individual settings. See OpenSSH man pages for more detailed information about OpenSSH daemon configuration.

- ssh-keygen, sftp, slogin, ssh, scp files to /usr/local/bin directory.

Configuring TCP Wrappers and the SSH Daemon

Create /etc/hosts.allow, and /etc/hosts.deny.
/etc/host.allow should contain what machines are allowed to access.

The following file gives ssh connection from the class C subnet.

```
#hosts within a class C network
sshd : 139.121.66.0/255.255.255.0
#x11 forwarding allowed for internal network
sshdfwd-x11: 139.121.66.0/255.255.255.0

#additional hosts from external network
sshd : 143.120.65.26
sshd : 143.120.65.27
sshdfwd-x11: 139.121.55.0/255.255.255.0
```

/etc/hosts.deny file looks like

```
# mail failed attempt to
ALL:ALL: /usr/bin/mailx -s "%s: connection attempt from %a"
root@localdomain.com
```

Then, edit the sshd_config file in /etc.

- Turn the X-forwarding on by changing X11Forwarding from no to yes.
- Disable Root Login over the network by changing PermitRootLogin from yes to no.
- Review the Appendix D, and verify that the settings are identical.
- Edit the line that starts with "Subsystem sftp… " to
  `"Subsystem sftp /usr/local/libexec/sftp-server"`

Note: From the potential ssh client machines, edit /etc/ssh/ssh_config (the client configuration file) and set the following parameters to the corresponding values.

ForwardAgent yes
ForwardX11 yes

Generate the server key and hostkey using ssh-keygen.

```
# ssh-keygen -t rsa1 -f /etc/ssh/ssh_host_key -N ""
# ssh-keygen -t rsa -f /etc/ssh/ssh_host_rsa_key -N ""
# ssh-keygen -t dsa -f /etc/ssh/ssh_host_dsa_key -N ""
```

Create a script (/etc/init.d/sshd) with the following content:

```
case "$1" in
'start')
        if [ -x /usr/local/sbin/sshd -a -f /etc/ssh/sshd_config ]; then
                /usr/local/sbin/sshd -f /etc/ssh/sshd_config

# sshd daemon is called with sshd_config file.  Make sure these files
# exist.

        fi
        ;;

'stop')
        kill 'cat /etc/ssh/sshd.pid'

# Make sure that sshd_config file has a line
# "PidFile /etc/ssh/sshd.pid" or
# the PidFile has to match the path to sshd.pid file

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

```
# ln -s /etc/init.d/sshd /etc/rc3.d/S70sshd
```

So that sshd will start at reboot.

To verify that sshd is listening,

```
# telnet localhost 22

Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
SSH-1.99-OpenSSH_3.0p1
```

Disable Inetd

At this point, network services such as telnet, ftp and other unnecessary network services are still enabled through inetd daemon. ssh has already been installed and configured with

TCP wrapper, so inetd service should be disabled by killing inetd process and renaming /etc/sbin/inetd binary to /etc/sbin/inetd.bak-and /etc/inetd.conf to /etc/inetd.conf.bak.

 # ps –ef | grep inetd
root **159** 1 0 Oct 16 ? 0:00 /usr/sbin/inetd –s

and

```
# kill -9 159
# mv /etc/sbin/inetd /etc/sbin/inetd.bak
# mv /etc/inetd.conf /etc/inetd.conf.bak
```

Edit /etc/init.d/inetsvc script, and comment out the following line that initiates inetd daemon.

```
# /usr/sbin/inetd -s &
```

Filesystem Permissions

When the operating system and the necessary software is installed, lock down the filesystem permissions to further tighten the security.

Solaris 8 contains many binaries with group-write bits set. A tool written by Casper Dik removes group and world-write permissions from the files listed in /var/sadm/install/contents.

Download fix-modes from ftp://ftp.fwi.uva.nl/pub/solaris/fix-modes.tar.gz.to the development system. Uncompress and compile the file, and copy "secure-modes" and "fix-modes" files to the target machine. Put them in /usr/local/fix-modes/, ancd restrict their permissions to 700 root:root.

Run fix-modes by typing the following:

 # sh fix-modes

For those files that are somehow not listed in that file,
run the following command to identify those files with group or world write permissions.
To find out if any files in the system contain group or world writable permissions, type

 # find / -type f \(-perm –u+w –o –perm –g+w \) –ls > /tmp/group_writable
and view the output file.

Also, take a note of all the setuid and setgid files.
 # find / -type f \(-perm –u+s –o –perm –g+s \) –ls > /tmp/seguid_setgid
and view the output file.

Set-uid, and set-gid files, especially when combined with group or world writable permissions, can become a serious security threat. Run the following command to identify the set-uid and world writable files.

```
# find . -type f -perm -4002 -ls
```

Remove set-uid and world-writable *permissions* immediately upon identification.

```
# chmod -s <filename>
# chmod -w <filename>
```

Testing and Backup

The production system is now set up. Connect this system to the network for testing purpose. If the test is unsatisfactory, remove the network connection, and resolve the problem.

Testing (from the production machine)

1. Reboot the machine. Verify that the console prompts for the password.

2. Verify that the changes made in Network Settings section are in effect.

e. g) Run the following
 # ndd /dev/arp arp_cleanup_interval
instead of
ndd *-set* /dev/arp arp_cleanup_interval *60000*
and verify that the settings are in effect.

Substitute the other devices (/dev/tcp, /dev/ip, etc) and parameters described in network settings section to verify that the settings are in effect.

3. Verify the following daemons are not running

rpcbind
cachefs
sendmail
inetd
automountd
nfsd

Example:
 # ps -ef | grep inetd
~~root 165 1 0 19:01:37 ? 0:00 /usr/sbin/inetd -s~~

4. Verify that syslogd is running from step 3.

5. Verify that the NFS, RPC, and inetd related ports are NOT open by running the following command (use –an option instead of –a to verify the port numbers):

```
# netstat -a

   * ftp     …              LISTEN
   * telnet        …        LISTEN
   * sunrpc        …        LISTEN
   * nfsd    …              LISTEN
   * time    …              LISTEN
   …
```

6. Verify that port 22 is open from the previous step. Port 22 is used by OpenSSH.

7. Verify that /usr and /opt partitions are not writable, and suid files do not work under /var, /usr, /opt, and /export/home directories.

From another machine

8. Verify that other ssh enabled machines can connect to this machine via ssh, sftp, and scp. The successful first connection will prompt for the confirmation that the server host key will be trusted.

9. Verify that this host can not be accessed via telnet, ftp, rlogin, rcp and rsh.

Backup

Upon successful testing, bring the machine down to the single user mode, mount fixed drives, and mount a tape drive. The backup prodecure is directly taken and applied from Solaris Security Step by Step v2.0.

```
# reboot -- -s
# fsck
# mount -a
# mt /dev/rmt/0 rewind
```

Identify all the partitions

```
# df -F ufs -n
```

```
# for dir in / /var /usr /opt /export/home
do
      /usr/sbin/ufsdump 0f /dev/rmt/0n $dir
done
mt /dev/rmt/0 rewoffl
```

Make at least two back-up copies, and store them separately in secure locations.

At this point, there should be reasonable protection implemented for this machine, and others that needs to make connection to this machine.

Maintenance

Now, the system should be reasonably secure. Periodically (daily), check http://sunsolve.sun.com to get the list of updated patches, and apply them if applicable. In addition, periodically back up the system, and consider options to analyze the backups (especially the root partition which is still writable) to verify the system integrity.

References

1. Solaris Operating Environment Security Updated for Solaris 8 Operating Environment, Alex Noordergraaf and Keith Watson, Sun BluePrints OnLine, April 2001

2. Solaris Operating Environment Minimization for Security: A Simple, Reproducible and Secure Application Installation Methodology Updated for Solaris 8 Operating Environment, Alex Noordergraaf, Sun BluePrints OnLine, November 2000

3. Solaris Operating Environment Network Settings for Security Updated for Solaris 8 Operating Environment, Keith Watson and Alex Noordergraaf, Sun Blueprints OnLine, December 2000

4. Solaris Security Step By Step v2.0, SANS Institute, 2001

5. http://www.sans.org/y2k/practical/Jeff_Campione_GCUX.htm
Solaris 8 Installation Checklist, Jeff Campione, 2000

6. http://www.usenix.org/sage/sysadmins/solaris/solaris/checklist.html
Hardening Solaris

7. Auditing in the Solaris 8 Operating Environment, William Osser and Alex Noordergraaf, Sun BluePrints OnLine – February 2001

8. http://www.cert.org/security-improvement/implementations/i041.08.html
Configuring and using syslogd to collect logging messages on systems running Solaris 2.X

9. http://www.mail-archive.com/openssl-announce@openssl.org/msg00024.html
Advisory describing PRNG (Pseudo Random number generator in OpenSSL.

10. http://lelandsystems.stanford.edu/services/ssh/sysadmin/
Installing Secure Shell (SSH) for Unix System Administrators

11. ftp://ftp.ca.openbsd.org/pub/OpenBSD/OpenSSH/portable/INSTALL
OpenSSH Installation Instruction

12. Building and Deploying OpenSSH for Solaris Operating Environment, Jason Reid and Keith Watson, Sun BluePrints OnLine, July 2001

13. Hacking Exposed: Network Security Secrets and Solutions, 3rd Edition, Joel Scambray, George Kurtz, Stuart McClure, September 27, 2001

14. SSH, The Secure Shell: The Definitive Guide,Daniel J. Barrett & Richard E. Silverman, O'Reilly and Associates Inc., 2001

Appendix A.

```
58 packages that can be removed from Solaris 8 Core
Installation

system      SUNWadmr       System & Network Administration
Root
system      SUNWatfsr      AutoFS, (Root)
system      SUNWatfsu      AutoFS, (Usr)
system      SUNWauda       Audio Applications
system      SUNWaudd       Audio Drivers
system      SUNWauddx      Audio Drivers (64-bit)
system      SUNWcg6        GX (cg6) Device Driver
system      SUNWcg6x       GX (cg6) Device Driver (64-bit)
system      SUNWdfb        Dumb Frame Buffer Device Drivers
system      SUNWdtcor      Solaris Desktop /usr/dt
filesystem anchor
system      SUNWfcip       Sun FCIP IP/ARP over
FibreChannel Device Driver
system      SUNWfcipx      Sun FCIP IP/ARP over
FibreChannel Device Driver (64 b it)
system      SUNWfcp        Sun FCP SCSI Device Driver
system      SUNWfcpx       Sun FCP SCSI Device Driver (64-
bit)
system      SUNWfctl       Sun Fibre Channel Transport
layer
system      SUNWfctlx      Sun Fibre Channel Transport
layer (64-bit)
system      SUNWftpr       FTP Server, (Root)
system      SUNWftpu       FTP Server, (Usr)
system      SUNWi15cs      X11 ISO8859-15 Codeset Support
system      SUNWi1cs       X11 ISO8859-1 Codeset Support
system      SUNWkey        Keyboard configuration tables
system      SUNWluxdx      Sun Enterprise Network Array sf
Device Driver (64-bit)
system      SUNWluxop      Sun Enterprise Network Array
firmware and utilities
system      SUNWluxox      Sun Enterprise Network Array
libraries (64-bit)
system      SUNWm64        M64 Graphics System
Software/Device Driver
system      SUNWm64x       M64 Graphics System
Software/Device Driver (64-bit)
system      SUNWmdi        Sun Multipath I/O Drivers
system      SUNWmdix       Sun Multipath I/O Drivers (64-
bit)
```

```
system      SUNWnamow      Northern America OW Support
system      SUNWnisr       Network Information System,
(Root)
system      SUNWnisu       Network Information System,
(Usr)
system      SUNWpcelx      3COM EtherLink III PCMCIA
Ethernet Driver
system      SUNWpcmci      PCMCIA Card Services, (Root)
system      SUNWpcmcu      PCMCIA Card Services, (Usr)
system      SUNWpcmcx      PCMCIA Card Services (64-bit)
system      SUNWpcmem      PCMCIA memory card driver
system      SUNWpcser      PCMCIA serial card driver
system      SUNWpl5u       Perl 5.005_03
system      SUNWpsdpr      PCMCIA ATA card driver
system      SUNWrmodu      Realmode Modules, (Usr)
system      SUNWses        SCSI Enclosure Services Device
Driver
system      SUNWsesx       SCSI Enclosure Services Device
Driver (64-bit)
system      SUNWsndmr      Sendmail root
system      SUNWsndmu      Sendmail user
system      SUNWsolnm      Solaris Naming Enabler
system      SUNWssad       SPARCstorage Array Drivers
system      SUNWssadx      SPARCstorage Array Drivers (64-
bit)
system      SUNWtleux      Thai Language Environment user
files (64-bit)
system      SUNWudf        Universal Disk Format 1.50,
(Usr)
system      SUNWudfr       Universal Disk Format 1.50
system      SUNWudfrx      Universal Disk Format 1.50 (64-
bit)
system      SUNWusb        USB Device Drivers
system      SUNWusbx       USB Device Drivers (64-bit)
system      SUNWwsr2       Solaris Product Registry & Web
Start runtime support
system      SUNWxwdv       X Windows System Window Drivers
system      SUNWxwdvx      X Windows System Window Drivers
(64-bit)
system      SUNWxwmod      OpenWindows kernel modules
system      SUNWxwmox      X Window System kernel modules
(64-bit)
```

Appendix B. /etc/default/sendmail

```
#
# Copyright (c) 2001 by Sun Microsystems, Inc.
# All rights reserved.
#
# $Id: sendmail,v 1.1 2001/03/04 07:26:16 kaw Exp $
#
# INTRODUCTION
#
#  This configuration file defines sendmail operating modes, queue
#  intervals, and any additional options.  Install this configuration
#  file to affect changes to sendmail operations.  For further
#  information regarding sendmail security, see the Sun Blueprints(tm)
#  OnLine article entitled "Solaris Operating Environment Security -
#  updated for 8".
#
#       http://www.sun.com/blueprints/0401/security-updt1.pdf
#
#  The latest version of this configuration file is available for the
#  Blueprints OnLine tools area at:
#
#       http://www.sun.com/blueprints/tools/
#
#  This configuration file only works for the Solaris 8 Operating
#  Environment release.
#
# INSTALLATION
#
#       # cp <file> /etc/default/sendmail
#       # chmod 644 /etc/default/sendmail
#       # chown root:sys /etc/default/sendmail
#
#  Keith A. Watson <keith.watson@sun.com>
#

#
# MODE
#
# This variable defines the mode in which sendmail will operate.  Only
# the background mode ("-bd") and the queue mode ("") make sense in
# this context.  sendmail will listen for SMTP connections and process
# queued mail in background mode.  In queue mode, it will ONLY process
# queued mail.
# The default mode is "-bd" if this variable is not defined.
#
MODE=""

#
# QUEUEUINTERVAL
#
# The queue interval defines how long sendmail waits before attempting
# to process any mail in the queue.  Use the following characters to
# define the time:
```

```
#
#       's' senconds
#       'm' minutes
#       'h' hours
#       'd' days
#       'w' weeks
#
# For example:
#       "15m" == 15 minutes
#       "1h"  == 1 hour
#       "3d"  == 3 days
#
# The default queue interval is "15m" (15 minutes) if this variable is
# not defined.
#
QUEUEINTERVAL="15m"

#
# OPTIONS
#
# This variable defines any additional options to sendmail.
# There is no default value for this variable.
#
OPTIONS=""
```

Appendix C. /usr/lib/newsyslog rotation script

```
#
#ident @Z%newsyslog     1.3     97/03/31 SMI
#
LOG=messages
cd /var/adm
test -f $LOG.2 && mv $LOG.2 $LOG.3
test -f $LOG.1 && mv $LOG.1 $LOG.2
test -f $LOG.0 && mv $LOG.0 $LOG.1
mv $LOG   $LOG.0
cp /dev/null $LOG
chmod 644    $LOG
#

LOG=syslog
if test -d $LOGDIR
then
        cd $LOGDIR
        if test -s $LOG
        then
                test -f $LOG.6 && mv $LOG.6  $LOG.7
                test -f $LOG.5 && mv $LOG.5  $LOG.6
                test -f $LOG.4 && mv $LOG.4  $LOG.5
                test -f $LOG.3 && mv $LOG.3  $LOG.4
                test -f $LOG.2 && mv $LOG.2  $LOG.3
                test -f $LOG.1 && mv $LOG.1  $LOG.2
                test -f $LOG.0 && mv $LOG.0  $LOG.1
                mv $LOG    $LOG.0
                cp /dev/null $LOG
                chmod 644    $LOG
                sleep 40
        fi
fi

LOG=authlog
if test -d $LOGDIR
then
        cd $LOGDIR
        if test -s $LOG
        then
                test -f $LOG.6 && mv $LOG.6  $LOG.7
                test -f $LOG.5 && mv $LOG.5  $LOG.6
                test -f $LOG.4 && mv $LOG.4  $LOG.5
                test -f $LOG.3 && mv $LOG.3  $LOG.4
                test -f $LOG.2 && mv $LOG.2  $LOG.3
```

```
                test -f $LOG.1 && mv $LOG.1  $LOG.2
                test -f $LOG.0 && mv $LOG.0  $LOG.1
                mv $LOG    $LOG.0
                cp /dev/null $LOG
                chmod 644    $LOG
                sleep 40
        fi
fi
LOG=loginlog
if test -d $LOGDIR
then
        cd $LOGDIR
        if test -s $LOG
        then
                test -f $LOG.6 && mv $LOG.6  $LOG.7
                test -f $LOG.5 && mv $LOG.5  $LOG.6
                test -f $LOG.4 && mv $LOG.4  $LOG.5
                test -f $LOG.3 && mv $LOG.3  $LOG.4
                test -f $LOG.2 && mv $LOG.2  $LOG.3
                test -f $LOG.1 && mv $LOG.1  $LOG.2
                test -f $LOG.0 && mv $LOG.0  $LOG.1
                mv $LOG    $LOG.0
                cp /dev/null $LOG
                chmod 644    $LOG
                sleep 40
        fi
fi

/usr/bin/chown -f root:sys /var/log/syslog*
/usr/bin/chmod -f 0640     /var/log/syslog*

/usr/bin/chown -f root:sys /var/log/authlog*
/usr/bin/chmod -f 0640     /var/log/authlog*

/usr/bin/chown -f root:sys /var/log/loginlog*
/usr/bin/chmod -f 0640     /var/log/loginlog*
```

Appendix D. SSH Configuration Files

D.1 sshd_config (Server Configuration)

```
# This is the sshd server system-wide configuration file.  See sshd(8)
# for more information.
Port 22

# PidFile is used to store the process ID for the OpenSSH daemon.
# The daemon is killed using the information in this file when
# the system shuts down
PidFile /etc/ssh/sshd.pid

Protocol 2,1

# Listen on all available addresses.
ListenAddress 0.0.0.0

# HostKey for protocol version 1
HostKey /etc/ssh/ssh_host_key

# HostKeys for protocol version 2
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_dsa_key

# The server uses another key
# for encrypting client/server communication
KeyRegenerationInterval 3600
ServerKeyBits 1024

# Use Authentication Logging facility
SyslogFacility AUTH
# Amount of information logged.
# Use DEBUG to maximize the amount of information
# being logged when troubleshooting
LogLevel INFO

# The user has 10 minutes to log-on.
LoginGraceTime 600

# Root user's ability to log in over the network should be
# disabled.
PermitRootLogin No

# Make sure that the
# user home directories, SSH configuration directory (~/.ssh),
# and the key files under SSH configuration directory are owned
# by the user or root, and is not group and writable.
# The user won't be able to log in if the above criteria are not
# satisfied when StrictModes is set to yes.
StrictModes yes

# enable RSAAuthentication and Public Key authentication
```

```
RSAAuthentication yes
PubkeyAuthentication yes
AuthorizedKeysFile      %h/.ssh/authorized_keys

# rhosts authentication should not be used
RhostsAuthentication no

# Don't read the user's ~/.rhosts and ~/.shosts files
IgnoreRhosts yes

# disable host based authentication using .(s)rhosts
# and /etc/(s)hosts.equiv all together.
RhostsRSAAuthentication no
HostbasedAuthentication no

# Uncomment if you don't trust ~/.ssh/known_hosts for
#IgnoreUserKnownHosts yes

# Enable to log in with the passwords /etc/passwd and /etc/shadow.
# over SSH tunnel.
PasswordAuthentication yes

# Root user should not be able to log in over the network.
PermitEmptyPasswords no

# Used for s/key. s/key is not used.
ChallengeResponseAuthentication no

# enable encrypted X tunnel for secure X.
X11Forwarding yes

# the display variable is set automatically by sshd
# It will start from 10.
X11DisplayOffset 10

# print /etc/motd file when the user logs on
PrintMotd yes

# Verify the connections are alive in every interval.
# Disconnect if the connection is bad.
# defined by the system TCP/IP configuration.
KeepAlive yes

# enable secure ftp
Subsystem       sftp    /usr/local/libexec/sftp-server
```

D.2 ssh_config (Client Configuration)

```
# This is ssh client systemwide configuration file.
# See ssh(1) for more information.
# This file provides defaults for users, and the values can
# be changed in per-user configuration files or on the command line.

# Enable forward agent and X11
    ForwardAgent yes
```

```
    ForwardX11 yes

# Authentication mechanisms. Do not use rhosts.
    RhostsAuthentication no
    RhostsRSAAuthentication no
    RSAAuthentication yes
    PasswordAuthentication yes

# When SSH fails, do not fall back on Rsh
    FallBackToRsh no
# Do not use rsh
    UseRsh no
#
    BatchMode no
# When connecting other machines over ssh
# ask the users if the unidentified or changed
# host key should be entered into the server database.
# Rely on the users to make the right decision for
# manageability of host databases.
    StrictHostKeyChecking ask
    CheckHostIP yes

# user identity and key files/databases/configuration
    IdentityFile ~/.ssh/identity
    IdentityFile ~/.ssh/id_dsa
    IdentityFile ~/.ssh/id_rsa
# ssh port
    Port 22
#
    Protocol 2,1
    Cipher blowfish
    EscapeChar ~
```