Securing Solaris

Configuration guidelines for implementing a secure Solaris installation

Allaire Security White Paper Series

(Version 1.0)

Abstract

Title	Securing Solaris
Date	January 8, 2001
Product	Solaris 2.6 and 7
Target Audience	Web Server Administrators
Abstract	Securing a server can be a difficult process, considering the vast number of security advisories an administrator must keep track of. This document and other lockdown documents are Allaire's effort toward making this job a little easier.

Allaire Corporation • One Riverside Center • 275 Grove Street • Newton • MA • 02466
www.allaire.com • info@allaire.com • (617) 219-2000 •
security issues: secure@allaire.com
document feedback: lockdown@neohapsis.com

Table of Contents

Solaris overview

This document will provide the reader with information on methods of securely implementing and locking-down Solaris-based systems. We assume that the reader already possesses knowledge of the Solaris installation process, configuration process and general system administration tasks.

Installation considerations

If you're fortunate enough to be installing Solaris from scratch, we have a few initial recommendations:

- Consider installing only 'Core System' support—this option installs the minimum amount of packages necessary to run Solaris. However, the 'core system' install will limit you to command-line administration because the graphical user interface (X Windows) is not installed with this option. There also may be other applications that will not work when only installing core system support. If you should choose to install the 'Developers System Support', you should remove all packages that are unnecessary, such as UUCP.

- Carefully select your partition sizes. While convenient, using the auto-layout tool typically causes headaches—it doesn't allocate enough space to critical partitions, particularly /var. A secure machine will have extensive system log capabilities enabled, so you need lots of log space. We recommend at least 400-500 MB of disk space for /var alone (auto-layout seems to allocate only 25-100 MB). Ideally, you should set your swap file to double the size of your system memory. At the very least, it should be the same size as your system memory. You can cheat by disabling multiple partitions: allocate your desired space to the swap partition and then use the rest for the / (root) partition. However, doing this breaks down filesystem segregation and should only be done on small disks where the threat of filling a particular partition is high. File system segregation is the preferred method since you can use mount options such as 'nosuid" while mounting the partition. It also helps to protect other filesystems from filling up because of a user's activities. More information can be found at http://www.sunworld.com/swol-10-1999/swol-10-security.html.

Installing patches

Security vulnerabilities are being discovered on a weekly basis. In response, vendors release full or partial upgrades and fixes to the newly discovered

security threat in the form of a “patch” or “hotfix.” Sun has also provided a ‘Patch Primer’ available at
http://sunsolve.sun.com/pub-cgi/show.pl?target=content/content1&ttl=article

It is crucial that organizations stay current with security alerts and fixes. In most cases, patches are the only way to close security holes in vendor-provided applications. Two good sources for security alerts are Security Focus (http://www.securityfocus.com/) and SAN’s Security Alert Consensus service (http://www.sans.org/newlook/digests/SAC.htm).

You can retrieve a list of applied patches by running ‘patchadd –p’ as root on your system.

Recommended patch cluster

Immediately after installation, install the recommended patch cluster. The patch cluster may contain kernel upgrades that require a system reboot. Upgrading the kernel early on will save you a reboot down the road (which may be an issue on mission-critical servers). You can ftp patches and patch clusters from ftp://sunsolve.sun.com/pub/patches. The recommended patch cluster will have a name similar to

```
<OS Release>_Recommended.(tar.Z or zip)
```

Important note: *If you installed core system support only, your patch-installation script will try using an application that was not installed, resulting in a patch installation failure. You can fix this by running the following commands:*

```
mkdir /usr/xpg4
mkdir /usr/xpg4/bin
ln –s /bin/grep /usr/xpg4/bin/grep
```

This will allow the patch installation script to use the default grep application. Sun has been notified of this problem, so it may be fixed in later versions.

Once downloaded, extract the cluster by issuing the following:

- For .zip extension (Solaris 7 & 8):

```
unzip 7_Recommended.zip
```

- For .Z extension:

```
uncompress <OS Release>_Recommended.tar.Z
tar xvpf <OS Release>_Recommended.tar
```

Once extracted, you must 'cd' to the created directory (which will be named "<OS Release>_Recommended") and run the patch installation script:

```
./install_cluster –nosave
```

Patch cluster installation errors

You may see installation errors while installing patches or patch clusters—this is not always a bad thing. Failure code '8' indicates the application to which the patch relates is not installed. Ignore these. Failure code '2' indicates the patch has already been installed. In both cases, there is little reason for concern. However, if you receive any other failure codes indicating the patch was not successfully installed, your network's security status may still be vulnerable. In this event, read the patch's included README file (this should be reviewed regardless) for system requirements and consult Sun's patch primer.

Stand-alone security patches

Unfortunately, Sun does not put all security-related patches in the recommended patch cluster, so reviewing the available patches and being aware of security-related issues is still necessary.

Frequently view the 'PatchReport' for your OS version release. It's available via ftp from: sunsolve.sun.com/pub/patches/Solaris<OS Release>.PatchReport. This report contains a section entitled "Patches Containing Security Fixes." Review every patch listed to see if it's applicable to your installation. When in doubt, download and attempt to install all of the listed security patches.

Accomplish package installation by using the 'patchadd' command. When you install a patch, a copy of the old (vulnerable) binaries are saved in the /var directory. We recommend you use the '-d' switch to keep patchadd from making copies of the vulnerable binary. However, by the using '-d' option you lose the ability to uninstall the patches.

To install a patch, uncompress/unzip the patch in a temporary directory, and then run:

```
patchadd (-d) <patch directory>
```

For example:

```
patchadd –d 106944-02
```

Note: *Older Solaris releases (pre-2.6) use 'installpatch' instead of 'patchadd'*

Be aware that installing patches can inadvertently turn on services that were previously disabled. Also, if you have replaced a standard Sun program with an

open source version (such as Sendmail), it may be overwritten. Ideally you want to test the patch on a non-production unit without using the '-d' option. After successful testing, apply the patch using '-d' to the production servers.

Disabling unused services

By default, Solaris installs and activates a wide range of frequently unneeded services. Most good security practitioners agree that you should only be running actively used services and subsystems. By disabling unneeded or unused services, you greatly reduce your system's potential vulnerabilities.

Services started on boot

The system should always start with the least amount of necessary services possible. Service initialization is controlled by the scripts found in the "/etc/rc*.d" directories. We will focus on /etc/rc2.d, which is the default run-level for Solaris.

In the /etc/rc2.d directory you'll see various scripts that begin with the letters 'K' or 'S'. Each script controls its service counterpart (i.e. S74syslog controls syslogd). All scripts beginning with the letter S are started at this run-level, while all scripts starting with the letter K are stopped ('K'illed). The number after the initial S or K specifies the order in which the scripts will be called.

Disabling scripts is as simple as renaming the specific 'S' entries to 'K' entries. For example, if you wanted to disable the Sendmail service, run:

```
mv S88sendmail K88sendmail
```

Renaming the script is preferable to deleting it since you can re-enable the service at boot simply by restoring the original name. Keep in mind, however, that while renaming the script stops the service from starting at boot, if the service was already running it continues running until the system is rebooted. You can stop services by calling the script with the 'stop' parameter before you rename it, for example:

```
./S88sendmail stop
```

Disabling services is entirely dependent on two things: what the system needs and for what it's being used. We recommend disabling the following, provided you're not using these services:

Note: *The numeric startup value may differ on your system. Also, installing updates and patches at a later date may recreate the 'S' scripts.*

NFS (Network File System) services - Disabling these will prevent your system from exporting and mounting NFS shares. NFS is a large security concern—more secure methods of remote filesystem sharing are available. Disable:

```
S28nfs.server
S73nfs.client
S73cachefs.daemon
S74autofs
S93cacheos.finish
```

Solaris auto-configuration services - These services provide a feature-set useful for recommissioning a server. Most installations will not need this feature so disable:

```
S30sysid.net
S71sysid.sys
S72autoinstall
```

RPC services - These have long been the source of many Solaris systems intrusions. RPC services primarily provide support for NIS/NIS+, remote status-querying and a few tools for messaging and bootp (diskless workstation) related services. In many cases RPC services are not necessary and won't be used. By disabling RPC services, administrators can close the door on a large array of vulnerabilities. Disable the following scripts:

```
S71rpc
S76nscd
```

Expreserve service - This handles the recovery/preservation of lost editor buffers of files being edited during a system crash. While expreserve is a useful feature, it has had numerous security problems. To disable expreserve, rename the following script:

```
S80PRESERVE
```

Sendmail - The use of Sendmail depends on whether or not your host must function as a mail server (this constitutes receiving incoming e-mail; outgoing e-mail is still possible without). If you aren't using Sendmail, disable it since it has a seemingly never-ending set of security flaws (although, to Sendmail's credit, no major problems have risen recently). If you disable Sendmail at boot you'll still be able to send outgoing e-mail (contrary to popular belief), however, Sendmail does require an extra entry be added to cron for this to work (discussed below). Script to disable:

```
S88sendmail
```

Other startup services/scripts include:

S01MOUNTFSYS
- mounts various disk partitions (required)

S20sysetup

- displays any trademark information (may be disabled)

S69inet & S72inetsvc
- configures the machine to use networking (required)

S74syslog
- the system log daemon (required)

S75cron
- task scheduler (required)

S99tsquantum
- configuration for Ultra-Enterprise 10000 (may be disabled if your system is not an UE10000)

S99audit
- handles audit logs generated by the kernel in response to the configured auditing policies (required)

S88utmpd
- maintains the utmp log, which tracks user logins, etc. (required)

S75savecore
manages core files that may be generated during system boot crashes (may be disabled, although not recommended). Don't forget that if this in enabled, you need to have enough primary swap space defined for the amount of memory in your system.

Network services started by inetd

Many services are started by requests from the network. A popular Unix service named inetd traditionally handles these requests, and starts up the proper service in response. The services inetd allows are specified within the /etc/inetd.conf configuration file. By default, Solaris has *many* services enabled. Rather than figuring out which of those many services to disable, start by disabling all of them and then enable only the ones you need. Luckily none of the services launched from inetd are required for core system operation (except for NIS/NIS+, which does depend on a few services made available from inetd).

You can disable a service by commenting out the appropriate command line in /etc/inetd.conf.

The common services launched from inetd include:

FTP - The FTP server lets your system act as a fileserver for FTP-capable clients. Most ISPs provide FTP functionality to their client base, letting them

upload material to hosted websites. FTP transmits passwords in plaintext—a serious security concern. If you can limit your users to using a more secure method of file transfer, such as SCP, shut off FTP. SCP is available in most SSH distributions. This is discussed in greater detail later in this document.

Telnet - Telnet is a classic service that lets you log into your system remotely. Like FTP, it also transmits passwords "in the clear." Unlike FTP, telnet access is an even greater security concern because it allows remote-shell access to your system. Consider turning telnet off and replacing it with something like secure-shell (SSH). SSH is discussed in greater detail later in this document.

Shell/login/exec (also known as rshd, rlogind and rexecd) - These are a suite of services that allow for both remote login and execution of commands. The same functionality is gained using SSH or telnet. These should be disabled unless you have applications specifically designed to use them—and even then, this is a security concern.

Fingerd - This lets attackers gain user (and possibly process) information. Fingerd should be disabled.

lpd (printer) services - This lets remote clients submit print jobs. Lpd has had numerous problems. If you're not using your host specifically as a print server, disable it.

TFTP - Similar to FTP, TFTP is a very simple file transfer service. Unlike FTP, TFTP uses *no* authentication. It's the source of numerous security problems and should always be disabled. If you require the use of TFTP for router images, boot images, etc.; manually enable it on a case-by-case basis.

Rusersd - Similar to finger, ruserd can be used to gain unauthorized user information. Disable it unless absolutely needed.

Rquotad - This is used to support UFS disk quotas for NFS clients. Disable it if you're not using NFS.

Sprayd - Sprayd is used for testing network connectivity. Disable it.

Walld - It allows remote posting of messages to users on the system. Disable it, unless needed.

Rstatd - This returns statistics about the system and should be disabled.

Rexd - Rexd allows for remote execution of commands. It is considered very insecure and should be disabled.

Cmsd - This is the calendar manager service that allows remote manipulation of calendar data. Cmsd has recently been the cause of many security vulnerabilities. Disable it.

Sadmind - Sadmind is part of the Solstice administration schema. If you're not using the Solstice administrator, disable sadmind.

Kerbd - This provides Kerberos authentication support for the system. If you're not using Kerberos, disable kerbd.

Cachefsd - Cachefsd offers filesystem support for NFS. If you're not using NFS, disable cachefsd.

Fs (fs.auto, the Sun font server) - This service is needed only on systems with the GUI (graphical user interface) installed. If you don't have the GUI installed, disable fs.

Ttdbserverd - This is a service for the ToolTalk database. Recently, ttdbserverd has been found to have security vulnerabilities and as such, it should be disabled.

Other legacy services to disable:
Comsat
Talk
UUCP
Name (not to be confused with named or the DNS protocol)
Systat
Netstat
Time

Secure, stand-alone systems usually have only HTTP and SSH enabled. Virtual hosting providers may need to provide FTP, as well.

Enable Logging

By default, Solaris does not direct many log/error messages to log files. Instead, Solaris is configured to send messages to the root console. These messages are very important as they frequently contain information on successful and failed login attempts, su usage, etc. If you don't change the default configuration, you may loose valuable information. To direct syslog to save this information to a file you must add the following line to your /etc/syslog.conf:

Note: *The white space in the following entry ****MUST***** be tabs.*

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

You must create the file /var/log/authlog if it doesn't already exist (Solaris 7 seems to have it already, although it is not setup to use it). You can do this by running the following commands (as root):

```
touch /var/log/authlog
chmod 600 /var/log/authlog
```

This will create the logfile, set 'root' as the owner and limit users' ability to view this file.

Another log file worth creating is 'loginlog,' which allows for the capture of failed login attempts (when someone does not provide a valid username/password combination). You can capture failed login information by running the following commands (as root):

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

Network configuration

The network drivers for Solaris are usually configured somewhat leniently. They are not optimized to stop denial of service attacks and may even let attackers 'bounce' attacks off the system into your private network. You can turn off specific features, such as IP forwarding and source routing support. This will make your system more resilient on the wire. We recommend placing the following commands in your /etc/init.d/inetinit file:

Note: *Place these commands at the end of the file. Placing them at the beginning may lead to them being changed further into the script.*

```
ndd –set /dev/ip ip_ire_flush_interval 120000
ndd –set /dev/ip ip_forwarding 0
ndd –set /dev/ip ip_strict_dst_multihoming 1
ndd –set /dev/ip ip_forward_directed_broadcasts 0
ndd –set /dev/ip ip_forward_src_routed 0
ndd –set /dev/ip ip_ignore_redirect 1
ndd –set /dev/ip ip_send_redirects 0
ndd –set /dev/arp arp_cleanup_interval 500
```

There is one more command to add, but it varies from one OS version to another. The command regulates how your system responds to certain denial of

service attacks. On Solaris 2.6, 7 (2.7) and 2.5.1 with patch 103582-12 add the line:

```
ndd –set /dev/tcp tcp_conn_req_max_q0 12288
```

On all other Solaris versions (prior to 2.6, except for 2.5.1 with the patch mentioned above) add:

```
ndd –set /dev/tcp tcp_conn_req_max 1024
```

Also, create the file '/etc/notrouter' which tells the system that this machine will not forward packets. To do this, add the lines:

```
touch /etc/notrouter
chmod 600 /etc/notrouter
```

Additional Service configurations

Sendmail

If you disabled Sendmail as suggested, but would still like to have e-mail delivered (outgoing) by your host you'll need to add a cron job or scheduled task. Adding the following line to the root crontab (by executing 'crontab –e' as root) enables outbound mail delivery:

```
0 * * * * /usr/lib/sendmail –q
```

This lets Sendmail attempt the delivery of queued (outgoing) e-mail every hour. E-mail is queued only after the initial delivery attempt fails. You can, of course, change the crontab values to run Sendmail at smaller or larger intervals.

Another option is to start Sendmail without the "-bd" option, which will make it a queue-watcher only. You should also turn off suid permissions.

If you choose to run sendmail to receive incoming e-mail, review and configure it appropriately. Sendmail configuration is beyond the scope of this document. You can review all of the documentation, along with configuration tips and advice at http://www.sendmail.org/ or http://www.sendmail.net/. O'Reilly also publishes a comprehensive guide to sendmail http://www.oreilly.com/

Cron

You may wish to restrict access to the cron service. Control this by placing a list of allowed users in the /etc/cron.d/cron.allow file. Users not specified in this file won't be able to submit cron jobs. This file may need to be created if it does not

already exist. You'll need to add root and user 'lp' if you're running printer services.

FTP

If you're running FTP, make sure you disallow FTP access to all the system accounts, as well as all other users not allowed to FTP files. You can disallow account access by placing the account name in /etc/ftpusers (you may need to create this file if it doesn't exist). This file contains a list of users, one per line, who are not allowed to FTP files. Minimally you should have root, daemon, listen, smtp, lp, bin, sys, nobody, noaccess, uucp, nobody4, nuucp and adm. Also, make sure this file is secure by running:

```
chmod 600 /etc/ftpusers
```

Additional considerations for Solaris 2.6 and later

Sun added a few security-related features in Solaris version 2.6 that should be considered:

Non-executable stack

Many of today's attack methods are based on a technique called "buffer overflowing" (see http://phrack.infonexus.com/search.phtml?view&article=p49-14). Sun has built specific checks into the base operating system to detect and prevent this type of attack. To enable this protection add the following two lines to the end of /etc/system:

```
set noexec_user_stack =1
set noexec_user_stack_log =1
```

This will stop most buffer-overflow based attacks and we highly recommend it.

Note: *you will have to reboot in order for the above configuration change to take effect. Be aware that this does not necessarily make you invulnerable, it only affords you a little added protection. More information can be found at* *http://www.L0pht.com/advisories/bufero.html* *or at* *http://www.sunworld.com/sunworldonline/swol-10-1999/swol-10-security.html*

Rhost authentication with PAM

PAM (Pluggable Authentication Modules) is an authentication system used by newer Solaris versions (2.6 and above). By default, it is configured to allow logins based on the insecure rhosts mechanism. Rhosts authentication allows a remote user access to a system based on the originating system (sometimes

referred to as a 'trust' relationship). For example, rhosts can be configured to automatically allow user@host1 to log into host2 without asking for a password—host2 'trusts' that user@host1 is, in fact, a valid user. If attackers gain assess to host1, they can use the trust relationship to log into host2 without any authentication. Therefore, disable (either remove or comment out) all lines in /etc/pam.conf that contain 'rhosts_auth.'

Enable strong TCP sequence numbers
All TCP connections start with a random number. If an attacker can predict what random number(s) your system will use, he or she can perform hijacking and spoofing related attacks. To avoid this, enable the stronger TCP sequence number generating algorithm by editing /etc/default/inetinit and changing TCP_STRONG_ISS to a value of 2.

Console security
Someone can sit down at the keyboard of your system and cause quite a bit of trouble. Minimize your risk by keeping your console in a physically secured area or a locked cabinet. In extreme circumstances you can disable the Stop-A keyboard sequence. To do this, add the following line to your /etc/default/kbd file:

```
KEYBOARD_ABORT=disable
```

Miscellaneous

SSH source
SSH (secure shell) is a secure replacement for telnet. SSH encrypts the entire session, and prevents passwords and data from being sent on the network in clear text. SSH packages usually include SCP (secure copy), which is a secure file transfer alternative to FTP.

SSH is available as a source package from many places. You can grab a copy from http://www.freessh.org/.

We recommend using version 1.2.27; starting with version 2.0, the licensing model becomes stricter.

Note: *Check the README file regarding corporate licensing. Your licensing setup may also require you to use the RSAREF library. Details about this library are included within the SSH source README.*

There is also an open-source version of SSH, called OpenSSH, available from http://www.openssh.com/. OpenSSH uses non-patented cryptographic protocols, which leads to less licensing restrictions. However, OpenSSH is not compatible with regular SSH (above), which uses patented cryptographic algorithms.

There are also commercial ssh packages available from http://www.datafellows.com/.

Compiling

Unfortunately, Sun does not provide the tools needed to compile the source SSH package. However, there are freeware tools available from the Solaris Freeware project at http://www.sunfreeware.com/. You can setup a working compiler by installing the 'gcc' package. Instructions on installing gcc are available at http://www.sunfreeware.com/ as well.

Note: *If you installed only 'Core System' support, you will be missing vital pieces needed to compile programs. You can still obtain these files by manually installing the following packages: SUNWbtool, SUNWsprot, SUNWtoo, SUNWhea, SUNWarc, SUNWlibm, SUNWlibms. More information is available at http://www.sunfreeware.com/faq.html and http://www.inscoe.org/load.cgi?compilesun.*

In a perfect world, you would install gcc and compile the SSH package on a non-production system with developer support and then move the finished binaries over to your production box (which should be only the minimal core support).

Commercial compilation tools are also available from Sun and other vendors.

NMAP

NMAP is a popular system service analysis tool, commonly referred to as a 'port scanner'. You can use nmap to determine what services are publicly being offered to the Internet. First you need to get NMAP from http://www.insecure.org/nmap/.

Once compiled and installed, you can invoke NMAP to scan your local system by running:

```
nmap localhost -p 1-65535
nmap localhost -sU -p 1-65535
```

This will look for and list all ports that are listening (labeled as 'Open') on your system. On a minimal, secure system, you would have:

```
Interesting ports on localhost (127.0.0.1):
Port    State       Protocol  Service
22      open        tcp       ssh
```

```
80      open        tcp       web
```

You might also have entries for telnet (port 23) and FTP (port 21). The second scan, which has the -sU parameter, tells NMAP to also check for open UDP ports as well. On a secure system, none should be found, which is indicated by a result message such as:

```
No ports open for host localhost (127.0.0.1)
```

If NMAP lists a port as 'open' for a service you are unfamiliar with, you should research into your system which application is using that port, and close it if the service is unnecessary.

Legal banners

You may want to include a legal warning of usage for the system when users log on. Your warning should be placed in /etc/issue (shown before users log in) and /etc/motd (shown after users log in).

Password management

You can set password aging (how often users are forced to change their passwords) by configuring the MAXWEEKS line found in /etc/default/passwd. You can also set the minimal password length with the PASSLENGTH command. We suggest a setting of PASSLENGTH=8.

TCP_Wrappers

Tcp wrappers is a security tool that allows you to control access to any particular application based on source IP addresses. This is frequently used with applications spawned out from inetd. The tcp_wrappers application distribution and documentation is available at ftp://ftp.porcupine.org/pub/security.

Further system lockdown

Alec Muffet, Wietse Venema, Casper Dik and many others have compiled comprehensive sets of shell scripts and programs into a package named 'Titan,' which was designed to secure and audit specific portions of Solaris. Minimally, we suggest running add-umask.sh, fix-modes.sh and file-own.sh. Also useful for web servers is wwwchk.sh. Titan is available at http://www.fish.com/titan/.

TripWire is another tool to help you in securing Solaris. TripWire is a file integrity tool. It can alert you as soon as a 'protected' file is tampered with. More information can be found at http://www.tripwire.com/.

Resources

Solaris Security FAQ:
http://www.sunworld.com/sunworldonline/common/security-faq.html