Global Information Assurance Certification Paper

SANS
Global Information Assurance Certification (GIAC)

GCUX
GIAC Certified Unix Security Administrator
Practical Assignment, Version 1.8, Option 1

Installing and Securing

Solaris 8 and WU-FTPD 2.6.1

Jennifer M. Harper
December 2001

Table of Contents

Introduction

This document will define a step-by-step procedure for installing, configuring, and securing an ftp server. This server will run WU-FTPD v. 2.6.1 under the Solaris 8 operating system. Section One of the document will detail the operating system configuration. Section Two will detail the installation, configuration, and securing of WU-FTPD v. 2.6.1. Section One is intended to provide a generic guide to securing Solaris 8 that could be followed regardless of what services the server would be ultimately providing.

The purpose of this server will be to allow the transfer of files between internal users and external collaborators. The scenario envisioned is that employees of the organization which owns the server need to make certain files available for individuals outside of the local network. Those outside the network also need to be able to return files to the internal users. The ftp server will be configured to allow guest logins for the external users to connect and retrieve outgoing files or upload incoming files, without needing real user accounts on the local system.

Document Conventions

The following typeface conventions are used in this document:

Typeface or symbol	Meaning	Examples
Times	Normal text	This is a sentence.
`Courier`	`Commands, file names, screen output`	Use `pwconv` to update the `/etc/shadow` file
`Bold`	Operator entered **`text, commands`** or **selected option**	Select **Networked** Modify `/etc/nsswitch.conf:` `hosts files` **`dns`**
Italic	Replace with *`actual information`*	`vi` *`filename`*
`n`	Replace with *`a number`*	`nameserver` *`nnn.nnn.nnn.nnn`*
#	Root user prompt	# vi /etc/passwd
`%`	Normal user prompt	`% ftp hostname.mydomain.com`

The following terms will be used throughout this document, and should be understood as defined here.

administrator/system administrator—the individual or group of individuals responsible for the configuration and maintenance of the server.

build machine—a machine separate from the server being configured, which has compilers, libraries and other build tools installed; connected to the network (see Network Environment for more details).

network administrator—the individual or group of individuals responsible for the configuration and support of the organization's network.

security policy—document that outlines the organization's guidelines and expectations for usage

of servers and network.

server/secure server—the machine being configured following the procedure in this document.

organization—the business or institution which owns the server and network being discussed.

Hardware

The hardware used for the purposes of this document was a Sun Ultra 60 workstation with a single Sparc processor, 512MB of RAM, and an 18 GB internal hard disk. However, the procedures are, for the most part, hardware independent. It will also be necessary to have some mechanism of transferring files to the secure server, without use of a network (see details, next section). For this example, an external DAT tape drive was used. Additionally, some non-network based backup device will be required to perform a baseline backup upon completion of the setup process, but before attaching the server to the network.

Network Environment

This document is intended to provide generally applicable guidelines for a Solaris 8 server. No particular assumptions are made about the network environment other than that a connection to the Internet will be available. For the purposes of this document all security is host based. In practice, it will be necessary for the system administrator to consult with the organization's network administrator in order to take into account the specifics of the network in which the server is being installed.

During the install process the server will be kept disconnected from the network, in order to protect it from possible compromise before the securing steps can be completed. The best way to do this is to simply physically disconnect all network cables from the server. As a result, any downloads that need to be performed during the install will be to an alternate machine. Additionally, the secure server will have as minimal an operating system installation as possible in order to perform its function as an ftp server. The secure server will not have any compilers or other build tools installed. This choice keeps the setup of the server as spartan as possible, and affords an additional level of security in that no unauthorized tools can be compiled directly on the server once it is in service. Ideally the system administrator should establish a single machine to be used as a build platform. This machine will have all necessary compilers installed and can be kept isolated from other machines on the network. Any applications discussed in the following sections which require compiling, must be built on this alternate machine. Only the final installations will be handled on the server being configured. Additionally, since the secure server is still disconnected from the network, some non-network based method of transferring the necessary files to the server must be established. Depending on available resources, media such as CDs or DAT tapes could be used for this purpose.

Location and Physical Security

Physical security is an important part of overall system security. If at all possible all Internet connected servers should be kept in a room designed specifically as a data center. This room should be kept locked, with keys and/or access codes given only to a small number of people. The room should also have climate control systems adequate for the equipment that will be kept

in the room, as well as a climate monitoring system which will alarm and notify the appropriate personnel if the conditions deteriorate.

Risk Analysis

Once the example server is set up it will be connected to the Internet to provide upload and download ftp services to a wide user community. There are several levels of risk inherent to this type of server. First, there are the concerns of any Internet connected server. The machine must be protected from unauthorized users and remote attacks. A major concern of any anonymous ftp server is that it not be used to transfer or store pirated software, illegally copied music files, "warez," etc. To provide some level of protection against this type of threat, truly anonymous logins will not be permitted to this server. Instead, guest access will be used. Guest ftp access is only somewhat more secure than anonymous. The guest user needs a login name and password, but this login is not intended to individually identify users. In order to protect the ftp server from misuse, the administrator will have to carefully configure WU-FTPD to limit the actions guest users can take, and the areas of the server they can access. Finally, the nature of the legitimate data transferred on the server must be considered. In the scenario envisioned for the server, it may be necessary to protect the data uploaded by one guest user from being accessed by other legitimate guest users. Guest users will be allowed to up- and download files from specified locations, but will not be able to browse the file structure.

Step-By-Step Guide

Section One: Operating System Setup

Boot from Solaris 8 CD

Insert the latest Solaris 8 CD in the server's CD and boot the machine. In this example the Solaris 8 10/00 media was used.

Format root disk

To prepare the system for installation, the installer needs to format the root disk, set up a mini-root to hold files during the installation process, and configure the swap partition.

```
format /dev/dsk/c0t0d0?    y
Enter a swap slice size between 352MB and 17263MB, default=512MB
```

Generally, the default swap size of 512 MB is fine. Note that it will not be possible to change the size of the swap partition later, so the size selected here must be appropriate for the final configuration for the server. For many years the rule of thumb for swap size was twice the physical memory. However, on modern systems a swap slice that large may be excessive.

```
Can the swap slice start at the beginning of the disk?     y
```

Install says it prefers to start the swap slice at the beginning of the disk. There is generally no reason to configure differently.

```
Is this okay?   y
```

Confirm the settings you have entered. The installer then formats the disk as specified, copies files to the mini-root and automatically reboots when complete.

Configuration options

During the install process, it will be necessary to set a number of configuration options. Some of the network options will vary depending on the environment. If you are unsure of any network configuration, consult your network administrator.

Select **Networked**

Though the machine is disconnected now, it will be connected to a network eventually.

Use DHCP **No**

Generally, a server would not use DHCP to obtain an IP address.

Hostname
IP address
Netmask

Enter the appropriate network information for the server.

Enable IPv6 **No**

Name service **None**

This server will use DNS for name service. However, select **None** during the installation. If DNS is selected here attempts will be made to contact the DNS server. Since this machine is disconnected from the network during the install, these connections will fail, causing problems during the installation. All the necessary configurations for DNS will be established in a later step.

Timezone
Date

Select the correct timezone for the server, verify that the date and time are approximately correct. Time synchronization will be configured in a later step to ensure the server maintains an accurate time.

Root password

Set a secure root password. This password should be chosen and documented in accordance with the security policy. Root passwords should not be written down where they could be found by unauthorized individuals. However, it is a good idea to document the passwords in case of emergency. Passwords recorded for this purpose should be kept in a locked safe with access provided only to those who need to know the root passwords.

Turn power management off
Don't ask; leave power management as set above

Since this machine will act as a server and needs to be available at all times, power management should remain off.

Network Proxy Configuration: Direct Connection to the Internet

Review and confirm what you have entered.

Install Solaris Software

Insert CD *Solaris 8 Software 1 of 2* when prompted.

Select Custom Install

Select Geographic Region and Locale

Select Software to be installed

De-Select any additional software that was automatically selected
WebStart Scan location **None**
64-bit support **Yes**
Core Solaris Software group

The goal is to install the minimum amount of software need to operate this server. The Core Solaris Software group contains most of the software needed, without a lot of unnecessary items. Any additional packages that are required may be installed later.

Layout FileSystem

The correct layout of filesystems has become a matter of much debate, and is ultimately a matter of the system administrator's preference. There is no particular "right" layout. A few points to keep in mind when determining the filesystem layout:

- ✓ The server will have problems if `/` or `/usr` run out of space, so those filesystems should be large enough to accommodate the operating system, plus have enough room from some growth. Most additional software should be installed in `/usr/local` or `/opt`. By placing `/usr/local` and `/opt` on separate filesystems from `/` and `/usr`, it can be ensured that `/` and `/usr` don't grow dramatically after the initial installation.
- ✓ Log files are stored on `/var`. When determining the size of this filesystem consider the level of logging this machine will require, as well as how long log files will need to be kept. The length of time the organization keeps log files should be outlined in the security policy.
- ✓ Consider the purpose of the machine. The bulk of the disk space will probably go to the particular application of this machine. For this exercise most of the 18 gig disk was put into the `/ftpdata` filesystem, which will provide the data storage area for the ftp server.

Select the root disk from the list of available disks and click **[Modify]**. Enter an appropriate size for each default filesystem, and add or change any other filesystems as needed. In this example the `/usr/local` filesystem was added as a separate system from `/usr`, `/home` was removed, and `/ftpdata` was added. Record the layout, as below.

slice	filesystem	Size (MB)
0	/	100
3	/usr	500
4	/usr/local	900

5	/var	2000
6	/opt	500
7	/ftpdata	14000

After finalizing the filesystem layout, click **[Install Now]**.

Reboot the system when installation is complete.

Configure Network

Once the initial operating system installation has completed, the next step is to configure the server's network settings.

Create an empty file `/etc/notrouter`, to prevent the machine from behaving as a router.[1]

```
touch  /etc/notrouter
```

Create the file `/etc/defaultrouter` containing the IP address of the gateway router for the network the server will reside on. If you are unsure of this number, check with your network administrator.

Create the file /etc/resolv.conf containing the DNS information for your network. Again if you are unsure of this information, check with the network administrator.

```
# vi /etc/resolv.conf

nameserver nnn.nnn.nnn.nnn
nameserver nnn.nnn.nnn.nnn
domain     mydomain.com
```

Add `dns` to `hosts` line in `/etc/nsswitch.conf`

```
# vi /etc/nsswitch.conf

hosts      files      dns
```

Install additional packages

You may wish to install select packages not included with the Core Software group. These packages can be found on the Solaris Software Installation CDs (1 and 2). For example, to install the packages required for NTP (Network Time Protocol):

mount CD[2]

```
# mount -r -F hsfs /dev/dsk/c0t6d0s0  /mnt
# cd /mnt/Solaris_8/Product
# pkgadd -d . SUNWntpr SUNWntpu
```

Install Recommended Patches

Since the release of any operating system installation CD functionality and security problems will

[1] Pomeranz, Hal ed., *Solaris Security Step by Step Version 2.0*, The SANS Institute, 2001, page 2
[2] Pomeranz, page 3

have been discovered. To fix these problems vendors release patches. To install the most current recommended patches from Sun, download the bundle from sunsolve.sun.com. Remember that the download will be done to the build machine, and the patch bundle will need to be relocated to the secure server prior to installation.

```
# ftp sunsolve.sun.com
      cd /patches
      bin
      get 8_Recommended.zip
# unzip -qq 8_Recommended.zip
# cd 8_Recommended
# ./install_cluster    -q -nosave
```

When the patch installation has completed remove the patch cluster and reboot to ensure that any kernel changes take effect.

Optimize Default Operating System Installation

Certain processes started by default at boot time are unnecessary or pose potential security risks. To disable the startup of these processes either remove or rename the scripts in the `/etc/rcN.d` directories. Only scripts with names beginning in a capital S will be run at boot time.

The following scripts should be removed.[3]

From `/etc/rc2.d`:

```
S30sysid.net
S71sysid.sys
S72autoinstall
S88sendmail
S73nfs.client
S71ldap.client
S71rpc
S80PRESERVE
S76nscd
S73cachefs.daemon
S93cachefs.finish
S74autofs
```

From `/etc/rc3.d`:

```
S15nfs.server
```

From `/etc/rcS.d`:

`S50devfsadm` (If not using hot-pluggable devices)[4]

Remove NFS configuration files files:[5]

```
/etc/auto_home
/etc/auto_master
/etc/autopush
/etc/dfs/dfstab
```

Modify `/etc/init.d/syslog` to cause `syslogd` to start with the `-t` option. This option

[3] Pomeranz, pages 6-7

[4] Raborn, Timothy E., *Installing and Securing Solaris 8*, http://www.sans.org/y2k/practical/Timothy_Raborn_GCUX.zip,2001, pg. 12.

[5] Pomeranz, page 12

prevents syslog from listening for traffic on UDP port 514. Only network loghost servers should listen for messages on this port.[6]

```
/usr/sbin/syslogd -t > /dev/msglog 2>&1 &
```

By default, the system will accept connections using a variety of inet services, listed in `/etc/inet/inetd.conf`. These services can be used to compromise security or perform denial of service attacks against the system. If the system is not going to perform any inet services, inetd can be completely disabled and the `inetd.conf` file removed. However, if the server will provide any inet services, as in the case of this FTP server, all unnecessary services must be removed from `/etc/inet/inetd.conf`. For the example system, the only remaining line in `inetd.conf` will be the ftp line, which will be further customized following the installation of TCP Wrappers and WU-FTPD, later in the process.

Modify filesystem mounting options in `/etc/vfstab` to be more secure. Mount `/usr` read-only, preventing any modifications to this filesystem. The root filesystem must be writable, but should be mounted with the logging option; the remount option is required for the other filesystem options to take effect. All other filesystems should be mounted `nosuid`.[7] The modified `/etc/vfstab` appears below. Note the changes in the `mount options` column.

```
# more /etc/vfstab
#device       device      mount       FS    fsck    mount   mount
#to mount   to fsck      point       type    pass    at boot options
#
#/dev/dsk/c1d0s2 /dev/rdsk/c1d0s2 /usr         ufs     1    yes     -
fd    -      /dev/fd fd      -      no   -
/proc -     /proc proc  -     no    -
/dev/dsk/c0t0d0s1      -     -     swap -      no   -
/dev/dsk/c0t0d0s0    /dev/rdsk/c0t0d0s0    /     ufs   1     no
     remount,logging
/dev/dsk/c0t0d0s3    /dev/rdsk/c0t0d0s3    /usr    ufs      1      no
     ro
/dev/dsk/c0t0d0s5    /dev/rdsk/c0t0d0s5    /var    ufs      1      no
     nosuid
/dev/dsk/c0t0d0s7    /dev/rdsk/c0t0d0s7    /home   ufs      2      yes
     nosuid
/dev/dsk/c0t0d0s6    /dev/rdsk/c0t0d0s6    /opt    ufs      2      yes
     nosuid
/dev/dsk/c0t0d0s4    /dev/rdsk/c0t0d0s4    /usr/local ufs   2     yes
     nosuid
swap  -     /tmp  tmpfs -     yes    -
```

Install the following `netconfig` script in `/etc/init.d`, and create a link `/etc/rc2.d/S69netconfig`

```
# vi /etc/init.d/netconfig[8]

     #!/sbin/sh

     ndd -set /dev/tcp tcp_conn_req_max_q0 8192
     ndd -set /dev/tcp tcp_ip_abort_cinterval 60000
```

[6] Pomeranz, page 9
[7] Pomeranz, page 13
[8] Pomeranz, page 10

```
      ndd -set /dev/ip ip_respond_to_timestamp 0
      ndd -set /dev/ip ip_respond_to_timestamp_broadcast 0
      ndd -set /dev/ip ip_respond_to_address_mask_broadcast 0
      ndd -set /dev/ip ip_ignore_redirect 1
      ndd -set /dev/ip ip_send_redirects 0
      ndd -set /dev/ip ip_forward_src_routed 0
      ndd -set /dev/ip ip_forward_directed_broadcasts 0
      ndd -set /dev/ip ip_forwarding 0
      ndd -set /dev/ip ip_strict_dst_multihoming 1
      ndd -set /dev/arp arp_cleanup_interval 60000
      ndd -set /dev/ip ip_ire_arp_interval 60000
# ln –s /etc/init.d/netconfig /etc/rc2.d/S69netconfig
```

Add the following lines to `/etc/system` to set some kernel variables to safer values, in order to prevent and log certain types of buffer overflow attacks.[9]

```
      * Attempt to prevent and log stack-smashing attacks
      set noexec_user_stack = 1
      set noexec_user_stack_log = 1
      * Set various parameters to more reasonable values
      set maxuprc = 128
      set sys:coredumpsize = 0
      * Require NFS clients to use privileged ports
      set nfssrv:nfs_portmon = 1
```

Reboot the system at this point to reconfigure the kernel with the new `/etc/system` configuration.

Configure System Logging

Add the following lines to `syslog.conf` to increase logging level, and separate log messages based on log facility.

```
      kern.info               /var/adm/kern.log
      user.info               /var/adm/user.log
      mail.info               /var/adm/mail.log
      daemon.info             /var/adm/daemon.log
      auth.info               /var/adm/auth.log
      syslog.info             /var/adm/syslog.log
```

Create log files in `/var/adm/` and set appropriate permissions.

```
# cd /var/adm
# for log in kern user mail daemon auth syslog
do
      touch $log.log
      chmod 600 $log.log
      chown root:sys $log.log
done
```

Create a `loginlog` to capture bad logins[10]

```
      # touch /var/adm/loginlog
```

[9] Pomeranz, page 11
[10] Pomeranz, page 14

All log files should be owned by root, and readable and writable only by root.

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

Log rotation

Configure some mechanism of log rotation. Logs should be rotated periodically to avoid running out of space on `/var`, and to keep individual logs to a manageable size. Rotated logs should also be purged periodically. The length of time log files are kept should be determined by the organization's security policy. Inconsistent handling of log files can be problematic should the logs ever be needed in a legal procedure.

The script below can be used to rotate logs. The logs will be copied to `/usr/local/logs`, named with a date stamp and compressed. Any logs in the destination directory older than 90 days will be deleted. The script can be run as a cron job daily, weekly, or monthly, as appropriate for the system. Before running this script, make sure that all the variables are correct for the local system, and that the `logdest` directory exists.

```
#!/bin/sh
logsrc='/var/adm/'
logdest='/usr/local/logs'
compress_directive='/bin/gzip -9 '
datestring=`date +%m%d%y`

for logname in auth.log kern.log syslog.log user.log daemon.log
mail.log loginlog
     do
cp $logsrc/$logname $logdest/$logname\.$datestring
cp /dev/null $logsrc/$logname
$compress_directive $logdest/$logname\.$datestring
     done
#prune old logs
find $logdest -mtime +90 -exec rm {} \;
```

Configure User Security

Unnecessary or insecure user accounts can be used to compromise the system.[11]
Remove the following unneeded users:

```
uucp
nuucp
lp
smtp
listen
nobody4
```

use `/dev/null` as shell for the following other system accounts to block logins or exploits of these accounts:

```
adm
daemon
bin
```

[11] Pomeranz, pages 16-17

```
nobody
noaccess
```

At this time create user accounts for any system administrators. Administrators should be required to log into their individual accounts and use `su` or `sudo` to gain root privileges.

Disable .rhosts

`Rlogin` and `rsh` can be a security risk and should be disabled by commenting or deleting the `rhosts_auth` line in `/etc/pam.conf`.[12]

Create empty files in root's home directory to attempt to protect against certain remote attacks.[13]

```
for file in /.rhosts /.shosts /.netrc /etc/hosts.equiv
do
      cp /dev/null $file
      chown root:root $file
      chmod 000 $file
done
```

Cron configuration

Only root should be allowed to run `cron` and `at` jobs. To configure this:[14]

```
# cd /etc/cron.d
# rm cron.deny at.deny
# vi cron.allow at.allow
```

Only root should be listed in the `cron.allow` and `at.allow` files.

Remove all `crontab` files other than root from `/var/spool/cron/crontabs`

Set permissions:

```
# chown root:root cron.allow at.allow
# chmod 400 cron.allow at.allow
```

Modify `/etc/default/cron` to enable logging of all cron activity.

```
CRONLOG=YES
```

Configure Warning Banners

Warning banners should be displayed whenever a user connects to the system. These banners should warn that only authorized use is permitted, that the system is monitored, and give an indication of how detected violations will be handled. The details of system usage standards should be outlined in the organization's security policy and which must be made available to all users, and agreed to when an account is issued. Due to the complex and ever-changing legal implications of system security statements, the exact nature and wording of the security policy, as well as the warning banners should be reviewed and approved by the organization's legal counsel.

By default, `/etc/issue`, which is a warning displayed before the login prompt, does not exist.

[12] Pomeranz page 17
[13] Pomeranz, page 17
[14] Pomeranz, page 18

It is important to provide a warning before login, so that the user has the opportunity to "turn back" if he does not agree to the conditions of logging in. The default `/etc/motd`, displayed after a successful login, advertises the OS version, and this information could be used to aid an attacker in exploiting known vulnerabilities specific to a particular OS. Some would advocate modifying the motd to advertise a false or non-existent OS version in attempt to thwart (or at least irritate) would-be intruders. However, it should be noted that this could also cause confusion for legitimate users, including other system administrators, within the organization, and is ultimately not *that* critical in system security. Whatever is done should be agreed upon as a site standard, and documented in the appropriate manner.

Install site standard `/etc/issue`. For example:

```
This system is for the use of authorized users only.

Individuals using this computer system without authority, or in
excess of their authority, are subject to having all of their
activities on this system monitored and recorded by system personnel.

In the course of monitoring individuals improperly using this system,
or in the course of system maintenance, the activities of authorized
users may also be monitored.

Anyone using this system expressly consents to such monitoring and is
advised that if such monitoring reveals possible evidence of criminal
activity, system personnel may provide the evidence of such
monitoring to law enforcement officials.

Logging on to this system indicates acceptance of the Acceptable Use
policies of Name of Organization.
```

In the `/etc/default` directory modify both `telnetd` and `ftpd` files to include the line[15]

```
BANNER="Authorized Users Only!  All access will be logged. \n"
```

In addition, `ftpd` should include the following line which will ensure that safe default permissions are set on up-loaded files.

```
UMASK=002
```

These changes should be made even if telnet and/or ftp connections to the system will not be allowed. This provides a certain fallback level of security.

Set the appropriate permissions on all warning banner files to protect them from tampering.

```
# cd /etc
# chown root:sys motd
# chown root:root issue
# chmod 644 motd issue
# cd /etc/default
# chown root:sys telnet ftpd
# chmod 444 telnet ftpd
```

Finally, enable and set the oem-banner to provide a warning to anyone who has gained physical access to the console. Again, this is a fallback level of security, and in many ways simply a legal technicality. If an individual has gained unauthorized physical access to the machine and has the

[15] Pomeranz, page 19

intent to compromise the system security, it is unlikely that a warning banner is going to scare him off. However, should the organization attempt to take disciplinary or legal action against the individual, it may be important to prove that all attempts to warn of the consequences of unauthorized access were made.

```
# eeprom oem-banner\?=true
# eeprom oem-banner=" Authorized Users Only!  All access will be
logged."[16]
```

Sendmail

By default all Solaris machines run sendmail in daemon mode. It is not necessary to run sendmail in this mode if the machine will not receive mail. The machine may occasionally need to send mail out, for example, to send automatic reports to an administrator, or for the convenience of a user. The can be accomplished with a minimal configuration. The automatic startup of sendmail was disabled earlier. To run sendmail in a minimal mode, install a basic sendmail.cf (see example below) and enter a cron job in root's crontab to periodically process the outgoing mail queue. The following line will process the queue once an hour. The frequency of queue processing can be increased or decreased depending on the circumstances of the machine.

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

Additionally, some system administrators may wish to replace the Sun version of sendmail with the open source version from sendmail.org. If this is done the administrator needs to be cautious when installing Solaris patches not to install any sendmail patches, as they will overwrite portions of the installed sendmail. In addition the administrator must keep current on patches released by Sendmail and security issues with the open source package. For the most part, unless the system will operate as a mail server, there is no reason to install the open source sendmail.

```
# Minimal Client sendmail.cf[17]

### Defined macros
# The name of the mail hub
DRmail.domain.com

# Define version
V8

# Whom errors should appear to be from
DnMailer-Daemon

# Formatting of the UNIX From line
DlFrom $g $d

# Separators
Do.:%@!^=/[]

# From of the sender's address
```

[16] Pomeranz, page 19
[17] Pomeranz, page 33

```
# Dq<$g>
Dq<$g>$|$g$.

# Spool directory
OQ/usr/spool/mqueue

### Mailer Delivery Agents
# Mailer to forward mail to the hub machine
Mhub,   P=[IPC], S=0, R=0, F=mDFMuCX, A=IPC $h
# Sendmail requires these, but they are not used
Mlocal, P=/dev/null, F=rlsDFMmnuP, S=0, R=0, A=/dev/null
Mprog, P=/dev/null, F=lsDFMeuP, S=0, R=0, A=/dev/null

### Rule sets - WHITESPACE BETWEEN  COLUMNS MUST BE TABS!!!

S0
R@$+            $#error $:      Missing user name
R$+             $#hub $@$R $:$1 forward to hub

S3
R$*<>$*         $n              handle <> error addresses
R$*<$*>$*       $2              basic RFC822 parsing
```

Configure Network Time Protocol (NTP)

Create the file `/etc/inet/ntp.conf` containing at least 2 network time servers:

```
server time1.mydomain.com
server time2.mydomain.com
driftfile /etc/inet/ntp.drift
```

Run Fix-Modes

`Fix-modes` is a tool, which modifies the ownership and permissions on various system files to be more secure. Download the source code from `ftp.wins.uva.nl/pub/solaris/fix-modes.tar.gz` to the build machine.

```
# gunzip fix-modes.tar.gz
# tar xf fix-modes.tar
# make CC=gcc
```

Transfer the source directory to the secure system and run the script:

```
# sh fix-modes
```

Install Third Party Applications

In addition to the core operating system, certain third party applications will be either required or extremely valuable in securing and/or administering the server. Remember that most of the steps to build the applications will be done on the build machine, and the compiled code moved to the secure server. Also note that any downloads mentioned would be to the build machine.

TCP Wrappers

TCP Wrappers can be used to provide increased security and logging capabilities for certain inet services.

Download the latest version of TCP Wrappers from `ftp.porcupine.org`. Note: for Solaris 8 the `-ipv6` version of TCP Wrappers is required.

```
# gunzip -c tcp_wrappers_7.6-ipv6.tar.gz | tar xf -
# cd tcp_wrappers_7.6-ipv6
```

Edit Makefile and set appropriate options

```
# vi Makefile
```

Uncomment the appropriate `REAL_DAEMON_DIR`. This is the directory where the actual inet daemons are installed. For Solaris 8 the standard location is `/usr/sbin`.

```
REAL_DAEMON_DIR=/usr/sbin
```

By default TCP Wrappers log to the `LOG_MAIL` facility. To avoid having these log messages interspersed with sendmail messages, change the `FACILITY` variable.

```
FACILITY=LOG_AUTH
```

Next build the binaries with the appropriate parameters.

```
# make sunos5 CC=gcc
```

Copy the binaries to the system and install.[18]

```
# mkdir -p /usr/local/sbin /usr/local/include /usr/local/lib
# for file in safe_finger tcpd tcpdchk tcpdmatch try-from
> do
> /usr/sbin/install -s -f /usr/local/sbin -m 0555 -u root -g
daemon $file
> done
# /usr/sbin/install -s -f /usr/local/include -m 0444 -u root -g
daemon tcpd.h
# /usr/sbin/install -s -f /usr/local/lib -m 0555 -u root -g
daemon libwrap.a
```

Configure TCP Wrappers

Create the file `/etc/hosts.deny` containing the line

```
ALL: ALL
```

This will ensure that all activity not expressly permitted will be denied.

Create the file `/etc/hosts.allow` containing lines for the services that will be allowed, and designating acceptable origins for the traffic. For this example ssh connections will be allowed from the local domain, and FTP connections will be allowed from anywhere.

```
# more /etc/hosts.allow
sshd: .mydomain.com
in.ftpd: ALL
```

Secure Shell (SSH)

It will be necessary to have some method of allowing remote connections to the server. In order to prevent traffic from being passed across the network in clear text ssh should be used. The ssh

[18] Pomeranz, page 23

installation has three components: zlib, OpenSSL, and OpenSSH. All three components follow a fairly typical, straightforward installation procedure.

Install zlib

Obtain the latest version of the zlib source code from ftp.freesoftware.com. Here version 1.1.3 was used.

Use the following steps to unpack, configure, build and install the software.

```
# gunzip -c zlib-1.1.3.tar.gz | tar xf -
# cd zlib-1.1.3
# sh configure
# make
# make install
```

Install OpenSSL

Obtain the latest version of the OpenSSL source code from ftp.openssl.org/source. Here version 0.9.6b was used.

Use the following steps to unpack, configure, build and install the software.

```
# gunzip -c openssl-0.9.6b.tar.gz | tar xf -
# cd openssl-0.9.6b
# sh config
# make
# make install
```

OpenSSH

Obtain the latest version of the OpenSSH source code from a site listed on http://www.openssh.com/portable.html. Here version 2.9p2 was used.

Use the following steps to unpack, configure, build and install the software. The flags given to `configure` specify the install location, and configure ssh to operate at a secure level commensurate with the general setup of the machine (i.e. take advantage of TCP Wrappers, etc.). The options used here are not necessarily ideal for every situation. Available options should be reviewed and selected as appropriate. See the OpenSSH `INSTALL` file for the complete list of configure options.

```
# gunzip -c openssh-2.9p2.tar.gz | tar xf -
# cd openssh-2.9p2
# sh configure -prefix=/usr/local -with-tcp-wrappers -without-
rsh -disable-suid-ssh[19]
# make
```

Copy the binaries and config files from the source directory to the secure server, and install in the correct locations under `/usr/local`. Check the `sshd_config` to verify expected locations.

Configure SSH

After installing ssh, a few configuration steps must be completed.

[19] Pomeranz, page 25

Generate host keys:[20]

```
ssh-keygen -b 1024 -N '' -f /etc/ssh_host_key
ssh-keygen -d  -N '' -f /etc/ssh_host_dsa_key
```

Install the following script in `/etc/init.d` Change any pathnames as necessary to match the local installation.

```
# sshd
#

case "$1" in
'start')
        if [ -f /usr/local/etc/sshd_config -a -f /usr/local/sbin/sshd
]; then
                echo "ssh starting."
                /usr/local/sbin/sshd
        fi
        ;;
'stop')
        [ ! -f /var/run/sshd.pid ] && exit 0
        sshpid=`cat /var/run/sshd.pid`
        if [ "$sshpid" -gt 0 ]; then
                echo "Stopping ssh."
                kill -15 $sshpid 2>&1 | /usr/bin/grep -v "no such
process"
        fi
        ;;
*)
        echo "Usage: /etc/init.d/ssh { start | stop }"
        ;;
esac
exit 0
```

Link the init script to `/etc/rc2.d/S75sshd` and `/etc/rc1.d/K11sshd`
Start ssh.

```
/etc/init.d/sshd start
```

Tripwire

Tripwire is a tool that will alert the system administrator when system files have been added or changed unexpectedly. Tripwire is freely available as an academic source release (ASR), or as a commercial version. The commercial version has many enhanced features and administrators may want to consider its advantages over the ASR version. However, for the purposes of this document, the freely available ASR version was used.

Download `Tripwire-1.3.1-1.tar.gz` from
http://www.tripwire.com/downloads/tripwire_asr.

Unpack the source

```
# gunzip -c Tripwire-1.3.1-1.tar.gz | tar xf -
```

Customize the `Makefile` and `config.h` file

[20] Pomeranz, page 26

```
# cd tw_ASR_1.3.1_src
# vi Makefile
DATADIR = /usr/local/bin/tw/databases
```

The data directory is where tripwire stores the file integrity checking databases. By default these databases are stored in `/var`.

Confirm that the variables `CC, LIBS, INSTALL,` and `HOSTNAME` are set appropriately for the system.

```
# vi include/config.h
```

set `#define DATABASE_PATH` to whatever `DATADIR` was set to in the `Makefile`

```
# make
# make tests
# make install
```

Customize `tw.config`

The default `tw.config` file is a good starting point, but will mostly not be ideal for any particular machine. Review the available configuration flags listed at the beginning of the `tw.config` file and customize the checks for the local system. The best way to fine tune `tw.config` is to run tripwire and examine the output. Look for reports of activity that *you* expected and modify `tw.config` so that *tripwire* expects it. Ideally tripwire should not produce any output unless something is wrong. Optimizing `tw.config` can be a time consuming process, but is worth the effort if it allows tripwire to reliably monitor the machine for unauthorized activity. Additionally, once the effort has been put in for one machine, it is likely that only minor changes will be needed in order to produce a `tw.config` file for other machines within the organization.

Initialize the database.

After the configuration has been set, tripwire must build a database of file information for the system. When tripwire is run normally the files present on the system will be checked against the information in the database to determine if a change has been made. To set up the database do:

```
# /usr/local/bin/tw/bin/tripwire –initialize
```

After the database is initialized it will need to be moved to the `databases` directory. (Tripwire will prompt you to do this.)

Run tripwire

In order to verify that your configuration is appropriate and the database is properly initialized by running tripwire from the command line. This initial run will produce some header information and should note the addition of the tripwire database to the system.

```
# /usr/local/bin/tw/bin/tripwire
```

If everything looks good, run tripwire with the –q flag. With this option tripwire will only produce a report if something has changed[21].

Once the configuration has been finalized, add a `cron` job to run tripwire regularly and alert the administrator when a change is noticed. Also remember that tripwire is only as reliable as the

[21] Pomeranz, Hal, *UNIX Security Tools*, Deer Run Associates, 2001, page 138

database. Ideally the database should be kept on read-only media. At a minimum, store a copy of the database on secure media, so that it can be compared to the local copy if a database compromise is suspected.

Section Two: Installing and Securing WU-FTP 2.6.1

Install WU-FTPD

Download the latest version of the wu-ftpd source code from ftp.wu-ftpd.org

```
./configure –prefix=/usr/local
make
make install
```

Configure WU-FTPD

See `doc/examples` in the WU-FTPD source directory for examples of available configurations options. Installing these files in `/etc` will activate their configurations. The `ftpusers` file, which lists accounts that are not permitted to connect via ftp should already be in place and include all system accounts. After installing WU-FTPD, verify that `ftpusers` has not been altered. The default `ftpconversions` file is also installed with WU-FTPD. For most purposes this file does not need to be modified.

A basic `ftpaccess` file was installed in `/etc`. For a more complete list of available options for this file refer to `doc/examples/ftpaccess.heavy`.

The desired configuration of the server is to not allow anonymous access, but allow guest access. To block anonymous logins, remove anonymous from the `class` line in `ftpaccess`. Since guest access will be permitted, `guest` should remain in the `class` line, and a `guestgroup` line must be added. In our example, the guest ftp user will be `guestftp` in group `ftp`. This account will be created in the next section.

```
class   all   real,guest
guestgroup ftp
```

To limit the actions that guest users can execute once connected to the ftp server, add the following lines to `ftpaccess`. Without these lines in `ftpaccess`, all options default to yes. Though anonymous use is not enabled, it does no harm to explicitly forbid various actions by anonymous users, and is again a fallback security measure. If guest users should be allowed to perform certain actions, remove guest from the relevant line. For example, one might consider allowing the guest user to overwrite files. In the example, overwrite ability poses an interesting dilemma. Since the guest user will not be able to see what files are on the server, an innocent user could inadvertently overwrite a file if this action is not prohibited. However, if it is prohibited, a malicious user who has some idea of how files are named could determine if a particular file is present by attempting to overwrite it. The administrator will need to consider the potential risks of various options and determine what is most appropriate for the local system.

```
delete      no    anonymous,guest       # delete permission?
overwrite   no    anonymous, guest      # overwrite permission?
rename      no    anonymous, guest      # rename permission?
chmod       no    anonymous,guest       # chmod permission?
```

```
umask       no    anonymous, guest        # umask permission?
```

In addition to regulating access using TCP Wrappers, it is possible to deny ftp access to certain hosts from within the `ftpaccess` configuration. The configuration also offers the option of displaying a customized denial message. To block access from hosts without resolvable IP address, use `!nameserved`.[22] The advantage to this method of denial is that the customized error message could inform the user of specifically why he was refused. For example, the following lines could be added to `ftpaccess`:

```
deny  *.evildomain.com      /usr/local/ftpmsgs/evil.denied
deny  !nameserved     /usr/local/ftpmsgs/noresolve
```

`/usr/local/ftpmsgs/evil.denied` could read:

```
No access is allowed from this domain.
```

while `/usr/local/ftpmsgs/noresolve` could read:

```
Sorry.  You have been denied access because your IP address
doesn't map back to a legitimate domain name.  Please contact
your local system administrator to fix this problem.[23]
```

Run WU-FTP

Modify `inetd.conf` to read:

```
ftp stream tcp nowait root /usr/sbin/tcpd in.ftpd -l -a
```

`-l` means to use syslog to log sessions
`-a` activates ftpaccess for access control

Recall that the ftp line should be the only line in `inetd.conf`

Restart `inetd`:

```
kill -HUP pid
```

Set Up Guest FTP

Create an entry for the guest user in `/etc/passwd` and `/etc/group`.

`/etc/passwd`:

```
ftpguest:*:101:50:FTP Guest User:/ftpdata/./:/bin/ftponly
```

Add `/bin/ftponly` to `/etc/shells`, or `ftpguest` will not be permitted to connect via ftp.

`/etc/group`:

```
ftp::50:ftpguest
```

Run `pwconv` to update `/etc/shadow` with the new account. Set a password for `ftpguest`.

When `ftpguest` connects WU-FTPD will `chroot()` into `/ftpdata`. From the perspective of the guest user `/ftpdata` will be `/`. In order for commands to work properly for the ftp user certain files and binaries must be placed in directories under `/ftpdata`, in the same

[22] Liu, Cricket, et al., *Managing Internet Information Services*, O'Reilly and Associates, Inc. 1994, page 77
[23] Liu, page 77

configuration they would normally be found under the root directory.

Build the necessary directory structure in the directory that the `chroot()` is done into. The directories should be execute-only, allowing the commands to run, but preventing the user from seeing the structure.

```
# cd /ftpdata
# for dir in etc usr usr/lib dev
do
     mkdir $dir
     chown root:daemon $dir
     chmod 111 $dir
done
```

Copy necessary files from the real directory structure to the ftpdata directory structure. The `ls` command (and its supporting libraries) is the most essential to install. Without a functioning `ls`, ftp will behave very strangely from the users point of view.

```
# cd /ftpdata
# cp -p /bin/ls bin/ls
# cp -p /etc/passwd etc/passwd
# cp -p /etc/group etc/group
# cp -p /usr/lib/ld.so usr/lib/
# cp -p /usr/lib/ld.so.1 usr/lib/

# cd /ftpdata/usr/lib
# for lib in libc.so libdl.so libintl.so libw.so
do
     cp -p /usr/lib $lib.1 .
     ln –s $lib.1 $lib
done
```

The `chroot'ed` ftp connection will require certain device files in the ftp root, in order to be able to establish network connections. The device files in `/ftpdata/dev` must be created with `mknod`. The syntax for `mknod` is:

```
     mknod name c|b major(n) minor(n)
```

where c or b indicates if the device is a character or block device. To find the correct major and minor numbers for the devices, list the files in `/dev/devices/pseudo/`.

```
# cd /dev/devices/pseudo
# ls -l *tcp *zero
crw-rw-rw-   1 root     sys       13, 12 Nov 29 11:35 mm@0:zero
crw-rw-rw-   1 root     sys       42,  0 Nov 29 11:35 tcp@0:tcp

# cd /ftpdata/dev
# mknod zero c 13 12
# mknod tcp c 40 0
```

Optionally, for on-the-fly compression and tar:[24]

```
     # cp /bin/gzip bin
     # cp /bin/tar bin
```

[24] Brennen, Michael, *How to setup WU-FTP Guest Accounts*, http://www.landfield.com/wu-ftpd/guest-howto.html, 1995, page 4

```
# chown root:bin bin/gzip
# chown root:bin bin/tar
# chmod 111 bin/gzip
# chmod 111 bin/tar
```

Modify files in `/ftpdata/etc`. The `passwd` and `group` files need to exist so that file ownership is correctly displayed within the `chroot'ed` environment. However, these files should be stripped down to contain only the necessary information, in order to avoid giving away too much information about the system and expose it to potential compromise.

Edit `passwd` to contain only necessary accounts, modified for safe ftp use[25]:

```
root:*:0:0::/:/etc/ftponly
ftpguest:*:101:50::/ftpguest/./:/bin/true
```

Edit `group` to contain only the following:

```
root::0:root
ftp::50:ftpguest
```

For additional security, create empty files in the ftp root[26]

```
cd /ftpdata
touch .rhosts .forward
chown root.root .rhosts .forward
chmod 400 .rhosts .forward
```

Create the directory structure for file transfers under `/ftpdata`. For this example only an incoming and an outgoing directory will be created. Set the permissions as appropriate. For the example, `ftpguest` will need to be able to write to `incoming`, but not read, so as to protect one user from seeing files uploaded by others. In this case, `ftpguest` will also not be able to see the files in `outgoing`. If the internal user wishes for an ftpguest user to download a file from `outgoing`, it will be necessary for the guest to know the exact file and path names.

Conclusion

At this point the ftp server should be secure and fully functional. As the final steps, reinitialize the tripwire database to include the changes made in Section Two and complete a baseline backup. Finally, attach the network cable and reboot the server. Once the server is up on the network, test the configuration, as outlined in the following sections.

Baseline Backup

Before connecting the server to the network, create a dump tape with a baseline backup of the secure system. The following script is an example of such a backup. The device paths and included filesystems should be changed where appropriate. Once the backup is completed, label and write protect the tape, and ensure that it is stored in an appropriate offsite location.

```
# bu0.sh
# script to do Level 0 backup

# set up environment
```

[25] Brennen, page 4
[26] Brennen, pages 4-5

```
DUMPC="/usr/sbin/ufsdump"
TAPE="/dev/rmt/0hn"
ROOT="/dev/dsk/c0t0d0s0"
USR="/dev/dsk/c0t0d0s3"
LOCAL="/dev/dsk/c0t0d0s4"
VAR="/dev/dsk/c0t0d0s5"
OPT="/dev/dsk/c0t0d0s6"
FTP="/dev/dsk/c0t0d0s7"

# make sure tape is rewound

mt -f $TAPE rewind

for fs in $ROOT $USR $VAR $OPT $LOCAL $FTP
do
        echo "Dumping: $fs"; echo
        $DUMPC 0ucf $TAPE $fs
        echo
done

# Stat tape drive

mt -f $TAPE stat

# Take tape offline

mt -f $TAPE offline

echo "Done"
date

exit 0
```

Check Configuration

An important final step in the configuration of the secure server is to confirm that the configuration works as expected. There are basically two aspects to this test: make sure the things that are supposed to work do, and the things that are not supposed to work do not.

- ✓ Attempt to connect using an enabled service from a domain or host denied access by TCP Wrappers. Confirm that the connection is denied and the activity is logged.

```
# tail auth.log
Dec 20 16:19:29 hostname sshd [621]: [ID 947420 auth.warning] refused
connect from sdn-ar-001pakoprP241.dialsprint.net
```

- ✓ Attempt to connect using a disabled service. Confirm that the connection is refused.

```
# telnet hostname
Trying nnn.nnn.nnn.nnn...
telnet: Unable to connect to remote host: Connection refused
```

- ✓ Connect to ftp server as a guest. Confirm that the connection is successful, upload and download work as expected, that the guest user sees only what he is supposed to in the file structure, and that all the activity is properly logged.

```
% ftp ftp.mydomain.com
Connected to ftp.mydomain.com
220 ftp.mydomain.com FTP (Version wu-2.6.1(1) Mon Dec 10 14:05:47 EST
2001) ready.
Name (ftp:username): ftpguest
331 Password required for ftpguest.
Password:
230 User ftpguest logged in.  Access restrictions apply.
ftp> pwd
257 "/" is current directory.
ftp> ls
200 PORT command successful.
550 No files found.
ftp> cd incoming
250 CWD command successful.
ftp> ls
200 PORT command successful.
550 Bad directory components
ftp> put file.coming.in
200 PORT command successful.
150 Opening ASCII mode data connection for file.coming.in.
226 Transfer complete.
local: file.coming.in remote: file.coming.in
707 bytes sent in 0.00079 seconds (872.86 Kbytes/s)
ftp> ls
200 PORT command successful.
550 Bad directory components
ftp> cd ..
ftp> cd outgoing
250 CWD command successful.
ftp> ls
200 PORT command successful.
550 Bad directory components
ftp> get file.for.guest
200 PORT command successful.
150 Opening ASCII mode data connection for file.for.guest (707
bytes).
226 Transfer complete.
ftp> put try.in.outgoing
200 PORT command successful.
553 try.in.outgoing: Permission denied.
ftp>
```

The above session shows that `ftpguest` is able to connect, and is `chroot'ed`. The system directories in the ftp root are not visible to the guest user. The guest is able to `cd` into either the `incoming` or `outgoing` directory, and can put files into `incoming`, retrieve files from `outgoing`, but cannot put files into `outgoing`.

Checking `WU-FTPD's xferlog,` as well as `auth.log` and `daemon.log` shows that the

connection and the activity was logged.

```
Mon Dec 10 14:20:31 2001 1 somehost.mydomain.com 675
/web/ftp/incoming/file.coming.in a _ i g ftpguest ftp 0 * c
Mon Dec 10 14:21:01 2001 1 somehost.mydomain.com 675
/web/ftp/outgoing/file.for.guest a _ o g ftpguest ftp 0 * c
```

- ✓ Attempt to connect to ftp server as a privileged user. Verify that the connection is refused and that the attempt is recorded by syslog.

```
Dec 20 15:43:16 hostname ftpd[580]: [ID 532633 daemon.notice] FTP
LOGIN REFUSED (username in /etc/ftpusers) FROM
otherhost.somedomain.com [nnn.nnn.nnn.nnn], root

Dec 20 15:43:27 hostname ftpd[580]: [ID 528697 daemon.info] FTP
session closed
```

- ✓ Check running processes. Due to the minimal installation that has been selected for this system, there will be very few running processes. In addition to the performance enhancements gained by this configuration, the limited processes allow an administrator to easily and quickly note if something is wrong, either due to unexpected processes or missing something that should be present.

```
# ps -ef
     UID   PID  PPID  C    STIME TTY      TIME CMD
    root     0     0  0   Dec 11 ?        0:00 sched
    root     1     0  0   Dec 11 ?        0:00 /etc/init -
    root     2     0  0   Dec 11 ?        0:00 pageout
    root     3     0  0   Dec 11 ?        0:42 fsflush
    root   215     1  0   Dec 11 ?        0:00 /usr/lib/saf/sac -t 300
    root   329     1  0   Dec 14 console  0:00 /usr/lib/saf/ttymon -g -h -p
hostname console login:  -T sun -d /dev/console -
    root   149     1  0   Dec 11 ?        0:00 /usr/sbin/syslogd -t
    root   202     1  0   Dec 11 ?        0:03 /usr/local/sbin/sshd
    root   141     1  0   Dec 11 ?        0:00 /usr/sbin/inetd -s
    root   150     1  0   Dec 11 ?        0:00 /usr/sbin/cron
    root   207     1  0   Dec 11 ?        0:00 /usr/lib/utmpd
    root   218   215  0   Dec 11 ?        0:00 /usr/lib/saf/ttymon
    root   574   556  0 14:04:34 pts/1    0:00 ps -ef
    root   556   554  0 13:41:23 pts/1    0:00 -sh
    root   554   202  0 13:41:17 ?        0:01 /usr/local/sbin/sshd
```

Ongoing Maintenance

Having the system is up and running is only the beginning. It is very important to continue to be vigilant and protect and verify the system. To that end, the following are examples of the tasks that should be performed on an ongoing basis.

Perform routine backups

The frequency with which system backups are done, as well as what files are backed up, and for how long backups are kept, will depend on the organization's policies, and the nature of the data stored on the server. Backups will be useful not only if data is lost, but in the event of a security compromise they can be used to restore the system to a known good state. In this sense, backups are particularly effective in conjunction with use of tripwire.

Run tripwire

Tripwire should be run periodically, and the output checked to verify that no unauthorized changes have made to the system. A cronjob should be added to run tripwire nightly, weekly, or monthly. The frequency with which tripwire is run should be determined by how long the organization can afford to potentially have the server run in a compromised state. Remember that running tripwire is useless if the report is not checked.

Immediately before making any modifications to the system run tripwire to verify that no unauthorized changes have been made. Also remember to update or reinitialized the tripwire database following any modifications.

Patch updates

It is important to stay on top of any OS patch releases, particularly patches which affect security. It is not necessary to install every patch as soon as it is released, and in fact, in some cases may be a mistake. However, it is a good idea to be aware of critical patch releases. Ideally, patches should be installed on a test system first and only installed on production systems after having been proven in the test environment.

In addition to OS patches, it is important to stay informed in regards to patches and known vulnerabilities in any third party software on the system, particularly when the third party application is critical to the functionality of the server, as is the case with WU-FTPD in this example.

Monitor log files

Though no one is likely to have time to carefully examine log files on a daily basis, it is a good idea to periodically spot check the logs for suspicious activity. On higher risk systems, (or all systems if resources allow), logs could also be sent to a central network loghost machine, or a printer for very high risk systems. Comparing these two copies of the log files for inconsistencies will alert the system administrator if log files have been altered—an indicator of possible system compromise. A log monitoring program such as `logcheck` or `swatch` should also be considered.

Document changes to the system

Keep a system log to keep a record of all modifications made to the server. The system log should begin with a description of the system, its purpose, and a completed checklist of the setup steps, as outlined in this document. Any future changes made to the system should be recorded in this log, noting the date, who performed the work, and what was done. This log file should be keep in a location accessible to all system administrators. A system log can be an invaluable resource in the event of a system failure or compromise, or if a new system administrator needs to take over administration of the server. These logs can also serve as learning tools within the organization, as well as an outline of the work being done by the system administration team.

Bibliography

Brennen, Michael, *How to setup WU-FTP Guest Accounts*, http://www.landfield.com/wu-ftpd/guest-howto.html, 1995

Liu, Cricket, et al., *Managing Internet Information Services*, O'Reilly and Associates, Inc. 1994

Pomeranz, Hal ed., *Solaris Security Step by Step Version 2.0*, The SANS Institute, 2001

Pomeranz, Hal, *UNIX Security Tools*, Deer Run Associates, 2001

Raborn, Timothy L., *Installing and Securing Solaris 8*, http://www.sans.org/y2k/practical/Timothy_Raborn_GCUX.zip, 2001