Solaris 8 Installation Checklist

This is the administrative checklist for installing Solaris 8 as securely as possible on a Sun SPARCengine UltraAXmp. The machine will be used as a file and compute server providing shell access to remote users accessing via the Internet, such as what might be used by an ISP. Users will not be able to compile software on this machine, but can execute their own shell scripts and run pre-compiled binaries which they move into personal data space. Users will receive mail on this host, but the host will not be used to relay mail for others. A DNS server will not be installed, but some notes are made below about particular changes one would make to a similar system used as a DNS server.

The host will be a DNS client, and will otherwise retrieve network services information from local files. Filesystems are local and not exported.

The hardware includes four Ultra II 300 MHz processors, 1 GB of memory, two hot-swappable UltraSCSI drives and a SCSI CD-ROM. (And of course, my first note about what one would do differently on a DNS server is to point out that you would use a lot less hardware.) No floppy drive should be attached. One of the disk drives will serve as the OS disk and the other will only be added when a backup is desired (thus the hot-swap feature). At several points in this exercise you will need to make use of another machine running Solaris 8 which has compilers installed.

OpenBoot Prom (OBP) Settings

From an **ok** prompt, make the following changes to secure OBP. The goal is to force the machine to boot to multi-user mode off of the OS disk unless a special *obp_password* which you choose is given. This password should NOT be the same as the root password. Moreover, if the password is forgotten, it cannot be recovered; you will not be able to boot intentionally to single-user mode or from a CD-ROM until a new EEPROM is obtained from Sun.

1. ___ setenv **security-mode** command
2. ___ setenv **security-password** *obp_password*
3. ___ setenv **auto-boot?** true (This is default, but you should make sure.)

If the machine is headless and you are using a serial console, you can make field diagnosis a little easier by ensuring that POST output is written to the console.

1. ___ setenv **diag-device** disk
2. ___ setenv **diag-switch?** true

Solaris Installation CD

Install Solaris from the latest Maintenance Upgrade release. This is the version of the OS with the most recent complete set of patches which have been tested together. However, it is not a fully-patched system. You will still need to apply the most recent recommended and security patch bundle.

1. ___________________________ Specify OS Version (*e.g.,* Solaris 8, 10/00).
2. ___ Set swap space equal to the amount of physical memory.
3. ___ Permit swap space to be placed at the beginning of the OS disk. NOTE: There is no inherent value in this, it just keeps you from having to know in advance the exact number of blocks the other filesystems will use. If you know your disk layout well, say no to this option and manually lay out the starting and ending blocks. If so, specify them here:

- **start**_____________________
- **end**_____________________

Once the mini-disk has been copied to the new swap partition, the machine will reboot and offer networking and basic host configuration options.

1. ___ Networked? (choose no).
2. ___________________________ Assigned hostname.
3. ___________________________ Assigned timezone.
4. ___ Give a Root Password (please don't write it in here).
5. ___ Power Management? (choose no)
6. ___ Power Management at Restart? (choose no)

Solaris Software 1 CD

Once the above questions have been answered, you will be prompted to insert the Solaris Software 1 of 2 CD. Once it is in the CD-ROM tray:

1. ___ Choose Custom Install.
2. ___ Choose appropriate region ________________ and locale ________________.
3. ___ Toggle all "Solaris 8 Documentation" choices **Off**.
4. ___ Toggle all "Solaris 8 Software 2 of 2" choices **Off**.
5. ___ Toggle all "Computer Systems Supplement CD" choices **Off**.
6. ___ Choose no additional software sources.
7. ___ Enable both 64-bit and 32-bit support.
8. ___ Choose "Core Solaris Software Group"

File System Layout

Do NOT use custom install. By default, Solaris will create very few partitions, and this will prevent you from locking down filesystems to be read-only or to prevent set-uid scripts. You should specify each of the following filesystems. Log below the size given to each and which slice corresponds to which filesystem. The */dev* and */devices* filesystems need only about 10MB each. Remember that the Sun disk image limits you to seven partitions per disk (and one is already being used by swap), so */* will need to be big enough to include */tmp*. Only use one of the two disk drives so that the other can be used for backup. (User files and home directories for a machine with this purpose should probably be kept on some attached storage device like a RAID array.)

1. Size of */* ___________ Slice ____________
2. Size of */dev* ___________ Slice ____________
3. Size of */devices* ___________ Slice ____________
4. Size of */opt* ___________ Slice ____________
5. Size of */usr* ___________ Slice ____________
6. Size of */var* ___________ Slice ____________

After the partitions are carved and software installed, you will be asked to insert the "Solaris 8 Software 2 of 2" CD. Choose "skip". The system will request a reboot.

Firmware Upgrades

The SPARCengine UltraAXmp should have adequate firmware to run Solaris 8 with 64-bit support. However, it is possible that a new OBP version will be necessary for future software enhancements. If you are prompted at reboot to upgrade firmware, do so.

1. ______ Did you need to upgrade the firmware? (y/n)
2. If so, what is the new OBP version? _______________ POST version? _______________

Additional Packages

There are additional packages not in the "Core Solaris Software Group" which you will want. This includes terminal information, system accounting utilities, Sun's NTP implementation, Berkeley-style binaries, the zlib compression library (for SSH), man pages and documentation. Since you're providing shell access, it would be kind to your users to provide some more powerful shells (like **tcsh** and **bash**), mail handlers like **pine** or the NMH tools, and it's always good to have **less**. Insert the "Solaris 8 Software 1 of 2" CD and install its packages.

1. ___ **/bin/mkdir /mnt/cdrom**
2. ___ **/etc/mount -r -F hsfs /dev/dsk/c0t6d0s0 /mnt/cdrom**
3. ___ **/bin/cd /mnt/cdrom/s0/Solaris_8/Product**
4. ___ **/usr/sbin/pkgadd -d . SUNWntp* SUNWlibC SUNWdoc SUNWscpu** (Say yes to all questions.)
5. ___ **/bin/cd /**

6. ___ **/etc/umount /mnt/cdrom**

Then replace that CD with "Solaris 8 Software 2 of 2" CD and install its packages.

1. ___ **/bin/mkdir /mnt/cdrom**
2. ___ **/etc/mount -r -F hsfs /dev/dsk/c0t6d0s0 /mnt/cdrom**
3. ___ **/bin/cd /mnt/cdrom/Solaris_8/Product**
4. ___ **/usr/sbin/pkgadd -d . SUNWter SUNWacc* SUNWman SUNWbash SUNWless SUNWtcsh SUNWzlib** (Say yes to all questions.)
5. ___ **/bin/cd /**
6. ___ **/etc/umount /mnt/cdrom**

If you have the "Software Companion" CD, insert it and install the SunFreeware packages. (Otherwise, you will have to compile the mail handlers manually on another machine and move them over here. You should also be aware that Sun provides no support or guarantees for the SunFreeware products, so if you are particularly paranoid, it would be advisable to compile these tools yourself.)

1. ___ **/bin/mkdir /mnt/cdrom**
2. ___ **/etc/mount -r -F hsfs /dev/dsk/c0t6d0s0 /mnt/cdrom**
3. ___ **/bin/cd /mnt/cdrom/components/sparc/Packages**
4. ___ **/usr/sbin/pkgadd -d . SFWpine SFWnmh** (Say yes to all questions.)
5. ___ **/bin/cd /**
6. ___ **/etc/umount /mnt/cdrom**

Sun Recommended and Security Patch Cluster

As mentioned above, the latest Solaris CD contains only the latest patches tested as a full set. There will undoubtedly be security and recommended patches for individual packages which have since been released. From a separate machine (this one is not networked and not yet secure enough to add to the network) which has the **md5sum** binary installed, fetch the cluster.

1. ___ Download *ftp://sunsolve.sun.com/pub/patches/8_Recommended.zip.*
2. ___ Download *ftp://sunsolve.sun.com/pub/patches/8_Recommended.README.*
3. ___ Download *ftp://sunsolve.sun.com/pub/patches/CHECKSUMS.*
4. ___ Run **md5sum 8_Recommended.zip** and verify that it matches what is listed in *CHECKSUMS*.
5. ___ Read *8_Recommended.README* for special instructions.

6. ___ Copy *8_Recommended.zip* to removable media which is readable by the target machine.

7. ___ Mount the removable media on the target machine.

8. ___ Copy *8_Recommended.zip* into */tmp* and **/bin/cd /tmp**

9. ___ Unzip *8_Recommended.zip* and execute its install script. Some patches will not install; these are for packages which were not installed.

Stopping Unnecessary Boot Processes

You only ever want to be in multi-user mode (with networking), single-user mode, or off, so you should remove startup scripts for all other run levels. You also remove scripts which start up insecure network services.

1. ___ **/bin/rm -f /etc/rc[013].d/***

2. ___ Rename autoconfiguration links */etc/rc2.d/{S30sysid.net,S71sysid.sys, S72autoinstall}* to */etc/rc2.d/{NOS30sysid.net,NOS71sysid.sys,NOS72autoinstall}*. This prevents someone with root access from executing **sys-unconfig** to wipe out all networking parameters.

3. ___ Rename NFS and cachefs links */etc/rc2.d/{K28nfs.server,S73nfs.client, S73cachefs.daemon,S93cacheos.finish}* to */etc/rc2.d/{NOK28nfs.server,NOS73nfs.client, NOS73cachefs.daemon,NOS93cacheos.finish,S74autofs}* to disable NFS mounts and exports.

4. ___ Rename RPC links */etc/rc2.d/{S71rpc,S76nscd}* to */etc/rc2.d/{NOS71rpc,NOS76nscd}* to disable RPC services.

5. ___ Rename expreserve link */etc/rc2.d/S80PRESERVE* to */etc/rc2.d/NOS80PRESERVE*. Expreserve recovers data from unsaved **vi** sessions, but historically has been vulnerable.

6. ___ **/bin/rm "/etc/auto_*" /etc/dfs/dfstab** (Be sure to note that there is no space between _ and * in this command!)

Next you should create a script which forces system daemons to start with a more secure **umask**:

1. ___ **/bin/touch /etc/init.d/umask.sh**

2. ___ **/bin/echo 'umask 022' > /etc/init.d/umask.sh**

3. ___ **/bin/chmod 744 /etc/init.d/umask.sh**

4. ___ **/bin/ln /etc/init.d/umask.sh /etc/rc0.d/S00umask.sh**

5. ___ **/bin/ln /etc/init.d/umask.sh /etc/rc1.d/S00umask.sh**

6. ___ **/bin/ln /etc/init.d/umask.sh /etc/rc2.d/S00umask.sh**

7. ___ **/bin/ln /etc/init.d/umask.sh /etc/rc3.d/S00umask.sh**

8. ___ **/bin/ln /etc/init.d/umask.sh /etc/rcS.d/S00umask.sh**

Tightening Networking

You should make the following adjustments to the **end** of */etc/init.d/inetinit* to protect the machine from SYN floods, ARP spoofs, smurf attacks and from being unwitting allies to DDoS attackers.

1. ___ Append to */etc/init.d/inetinit*: **ndd -set /dev/tcp tcp_conn_req_max_q0 10240**.
2. ___ Append to */etc/init.d/inetinit*: **ndd -set /dev/ip ip_ignore_redirect 1**.
3. ___ Append to */etc/init.d/inetinit*: **ndd -set /dev/ip ip_send_redirects 0**.
4. ___ Append to */etc/init.d/inetinit*: **ndd -set /dev/ip ip_ire_flush_interval 60000**.
5. ___ Append to */etc/init.d/inetinit*: **ndd -set /dev/arp arp_cleanup_interval 60**.
6. ___ Append to */etc/init.d/inetinit*: **ndd -set /dev/ip ip_forward_directed_broadcasts 0**.
7. ___ Append to */etc/init.d/inetinit*: **ndd -set /dev/ip ip_forward_src_routed 0**.
8. ___ Append to */etc/init.d/inetinit*: **ndd -set /dev/ip ip_forwarding 0**.
9. ___ Append to */etc/init.d/inetinit*: **ndd -set /dev/ip ip_strict_dst_multihoming 1**.

Next you should modify */etc/init.d/inetsvc* to disable DHCP and inetd. (This will eliminate remote access until **ssh** is installed below.) There is only one section you need to keep: the ifconfig line which resets netmask and broadcast addresses. Comment out every other line in the file except the one which says:

- **/usr/sbin/ifconfig -auD4 netmask + broadcast +**

NOTE: If you were to use this machine as a DNS server, you should also uncomment the conditional which asks whether **/usr/sbin/in.named** and **/etc/named.conf** exist, and if so, to start up **in.named**. (However, you should also get the latest copy of BIND from ISC and replace Sun's binary with one you compiled.)

Since **inetd** has been disabled, remove its configuration file. A later reappearance will be a smoking gun.

1. ___ **/bin/rm /etc/inet/inetd.conf /etc/inetd.conf**

File System Configuration

There are two levels of protection which you want to enforce here. First, you want to prevent trojan horse programs from replacing system binaries in */usr* and */opt*. So you mount them read-only. Second, you want to prevent set-uid scripts from executing on any of these filesystems. You made this possible by creating */dev* and */devices* filesystems above. Mounting a filesystem **nosuid** also prevents devices from operating, so you need those to be separate filesystems from the root directory in order to mount */* **nosuid**. In */etc/vfstab*:

1. ___ Change the **rw** option in the last column of the */usr* entry to **ro**.

2. ___ Change the **rw** option in the last column of the */opt* entry to **nosuid,ro**.

3. ___ Change the **rw** option in the last column of the */var* entry to **nosuid**.

4. ___ Change the **rw** option in the last column of the */* entry to **remount,nosuid**.

NOTE: If you were running this machine as a DNS server, you might leave */var* mounted **rw**. This would permit you to create a chrooted environment for BIND somewhere under the */var* directory, since a chrooted environment requires devices. Given this particular file system design, */var* is likely the best candidate for being mounted **rw** since it generally should not contain executables. However, it is far from a perfect choice because it certainly can contain them.

You should also ensure that removable media do not include set-uid executables. Since this machine is kept in a data center away from users, Volume Management should be turned off.

1. ___ **/usr/sbin/pkgrm SUNWvolr SUNWvolu SUNWvolg**

Administrative Accounts

Several of the accounts in */etc/passwd* are unneccessary, but to ensure that older programs do not break, you will just effectively disable them. You also should ensure that these accounts cannot use **ftp**, **cron** or **at**.

1. ___ Make */dev/null* the default shell for all users other than root or sys in */etc/passwd*.

2. ___ Make */sbin/sh* the default shell for root and sys.

3. ___ Issue **/bin/passwd -l <user>** for every user in */etc/passwd* other than root. This will lock out the accounts. (Replaces "NP" in the *shadow* file with "*LK*".)

4. ___ Remove crontab entries in */var/spool/cron/crontabs* for all users except root and sys.

5. ___ Add adm, lp, uucp and nobody4 to */usr/lib/cron/at.deny*.

6. ___ Add adm, lp, uucp and nobody4 to */usr/lib/cron/cron.deny*.

7. ___ Create an */etc/ftpusers* file containing all users in */etc/passwd*.

8. ___ **/bin/chown root:root /etc/ftpusers**

9. ___ **/bin/chmod 600 /etc/ftpusers**

Remaining Network Services

Not many. But the host needs to be a DNS client and should reference local files for password and group information. Let IPADDRESS be the IP address of the trusted DNS server, and GATEWAY be the IP address of the default router.

1. ____________________ IPADDRESS?

2. ___ **/bin/touch /etc/resolv.conf**

3. ___ **/bin/echo 'nameserver <IPADDRESS>' > /etc/resolv.conf**
4. ___ **/bin/chown root:root /etc/resolv.conf**
5. ___ **/bin/chmod 644 /etc/resolv.conf**
6. ___ Set every entry in */etc/nsswitch.conf* to be "files", except "**hosts: files dns**".
7. ____________________ GATEWAY?
8. ____ **/bin/touch /etc/defaultrouter**
9. ____ **/bin/echo '<GATEWAY>' > /etc/defaultrouter**

Improved Logging

You should ensure that security-related events are logged by **syslog**. This includes reboots, failed login attempts and all **su** attempts.

1. ___ Append to */etc/syslog.conf*: **auth.info	/var/log/authlog** (Must be TAB separated.)
2. ___ **/bin/touch /var/log/authlog**
3. ___ **/bin/chown root:sys /var/log/authlog**
4. ___ **/bin/chmod 600 /var/log/authlog**
5. ___ **/bin/touch /var/log/loginlog**
6. ___ **/bin/chown root:sys /var/log/loginlog**
7. ___ **/bin/chmod 600 /var/log/loginlog**

Next you should create a log rotation script for these files. The easiest way to do this is to add lines to the existing */usr/log/newsyslog* script.

1. ___ In the syslog section of Solaris 8's default */usr/log/newsyslog* (after messages), LOGDIR and LOG are defined, LOGDIR is tested, and then LOG is tested. Move the **LOG=syslog** line inside the first conditional, after **cd $LOGDIR** and before **if test -s $LOG**.
2. ___ Now you have an atomic section from **LOG=syslog** down to **fi** which names a particular log, tests if it exists, and if so rotates it. Copy this entire section twice, so that after **if test -d $LOGDIR** it appears three times.
3. ___ There should now be three lines which say **LOG=syslog**. Keep the first one as is.
4. ___ Change the second one to say **LOG=authlog**.
5. ___ Change the third one to say **LOG=loginlog**.
6. ___ Save the file and exit.

7. ___ **kill -HUP `/bin/cat /etc/syslog.pid`**

System and Process Accounting

Earlier you installed the Solaris system accounting packages, and you made sure that the sys user had a viable path so that system accounting could function. To get it started:

1. ___ Uncomment the two conditionals in */etc/init.d/perf*.
2. ___ Uncomment the **sa1** and **sa2** cron entries in */var/spool/cron/crontabs/sys.*

Process accounting can take up a large amount of space on the drive and will cause performance degredation. However, it provides some excellent information about every process running. To activate and look at its output:

1. ___ Turn accounting on and write the output to */var/adm/pacct*: **/usr/lib/acct/accton /var/adm/pacct**
2. ___ View the contents of */var/adm/pacct*: **/bin/acctcom**
3. ___ Turn accounting off: **/usr/lib/acct/accton**

If this information is desired, you should set up a script to periodically zip up this output and rotate the log. You can use the syslog template above and then cron your script.

Auditing in the Solaris Basic Security Module

Solaris' Basic Security Module provides thorough auditing capabilities, to the point where every action taken by any user on the system can be audited. You should be very careful, however, with what signals you want captured, because you can run out of disk space very quickly. To start some basic auditing:

1. ___ Create */etc/security/audit_startup* which looks like:

```
#!/bin/sh
/usr/sbin/auditconfig -conf
/usr/sbin/auditconfig -setpolicy none
/usr/sbin/auditconfig -setpolicy +cnt
/usr/sbin/auditconfig -setpolicy +argv
```

2. ___ Create */etc/security/audit_control* which looks like:

```
dir:/var/audit
flags:fw,fm,fc,fd,lo
minfree:10
naflags:lo
```

3. ___ **/etc/init.d/audit start**

This will audit file writes, file modifications, file creations, file deletions and login/logouts. If you have the disk space and want to expand the audit, **man audit_control**. You should also consider rotating this audit results (and compressing older versions) using the syslog rotation above as a template.

Sendmail

Before you even begin to set up sendmail, ensure that the DNS server you trust specifies this server as a recipient of mail for your domain.

Sun's sendmail binary should be replaced with one built from source. You will need to do this on a separate host which has compilers installed and then move the binaries and libraries over to this machine. From the other machine:

1. ___ Download *ftp://ftp.sendmail.org/pub/sendmail/sendmail-8.9.3.tar.gz*
2. ___ Unpack: **gzip -dc sendmail-8.9.3.tar.gz | tar xvf -**
3. ___ **cd sendmail-8.9.3/BuildTools/OS**
4. ___ In the *SunOS.5.8* file, change the line **define(`confENVDEF', `-DSOLARIS=20800 ')** to **define(`confENVDEF', `-DSOLARIS=20800 -DUSE_VENDOR_CF_PATH')**
5. ___ **cd ../../src**
6. ___ **sh Build**
7. ___ **cd obj.SunOS.5.8.sun4** and save the **sendmail** file in that directory onto removable media which is readable by the target machine.
8. ___ **cd ../../cf**
9. ___ **mkdir local_config ; cd local_config**
10. ___ You want your target machine to accept and forward mail, but you want to prevent spam, you don't want to be a relay host, and you want to avoid aliases which try to run shells. Let DOMAINNAME be the name of your domain. Edit *sendmail.mc* to look like this:

```
include(`../m4/cf.m4')
OSTYPE(`solaris2')

define(`confMAX_HOP',`25')
define(`LOCAL_SHELL_PATH', `/dev/null')
define(`confPRIVACY_FLAGS',`noexpn,novrfy')
define(`confSMTP_LOGIN_MSG', `$j mailer ready at $b')
define(`confMIME_FORMAT_ERRORS',`false')
FEATURE(use_cw_file)

MAILER(smtp)
MASQUERADE_AS(<DOMAINNAME>)
```

11. ___ **m4 sendmail.mc > sendmail.cf**
12. ___ **echo <DOMAINNAME> > sendmail.cw**

13. ___ Save **sendmail.cf** and **sendmail.cw** onto the same removable media on which you put the sendmail binary.

Now you should move the new sendmail files to the target host:

1. ___ Mount the removable media on the target machine.
2. ___ Copy **sendmail** into */usr/lib*, replacing Sun's default copy.
3. ___ **/bin/chmod 6551 /usr/lib/sendmail**
4. ___ **/bin/chown root:root /usr/lib/sendmail**
5. ___ Copy **sendmail.cf** and **sendmail.cw** into */etc/mail*, replacing Sun's default copies
6. ___ **/bin/chown root:root /etc/mail/sendmail.c***
7. ___ **/bin/chmod 644 /etc/mail/sendmail.c***
8. ___ **/etc/init.d/sendmail stop**
9. ___ **/etc/init.d/sendmail start**

Installing TCP Wrappers

You want to force all remote connections to this machine to be made via SSH and through TCP Wrappers. Once again you will need to build from the source on a separate host which has compilers installed and then move the binaries and libraries over to this machine. From the other machine:

1. ___ Download *ftp://ftp.porcupine.org/pub/security/tcp_wrappers_7.6.tar.gz*
2. ___ Unpack: **gzip -dc tcp_wrappers_7.6.tar.gz | tar xvf -**
3. ___ **cd tcp_wrappers_7.6 ; chmod 644 Makefile**
4. ___ Edit **Makefile** by uncommenting the instance of **REAL_DAEMON_DIR** which appears after the comment **# SysV.4 Solaris 2.x OSF AIX**. Also set **FACILITY = LOG_AUTH**.
5. ___ **make sunos5**

After TCP Wrappers builds, you can move it from the compiler host to the target host:

1. ___ On the compiler host, save the files **safe_finger**, **tcpd**, **tcpdchk**, **tcpdmatch**, **try-from** , **tcpd.h** and **libwrap.a** onto removable media which is readable by the target machine.
2. ___ Mount the removable media on the target machine.
3. ___ On the target host: **mkdir /usr/local/sbin /usr/local/include /usr/local/lib**
4. ___ Copy **safe_finger**, **tcpd**, **tcpdchk**, **tcpdmatch** and **try-from** from the removable

drive to */usr/local/sbin*. Set each of their permissions to 0555. Make root the owner and daemon the group.

5. ___ Copy **tcpd.h** from the removable drive to */usr/local/include*. Set its permissions to 0444. Make root the owner and daemon the group.

6. ___ Copy **libwrap.a** from the removable drive to */usr/local/lib*. Set its permissions to 0555. Make root the owner and daemon the group.

Installing OpenSSH

Now that TCP Wrappers is built, you can build SSH with TCP Wrappers support. The instructions here show you how to build OpenSSH, which is ported to other Unices from the SSH implementation in OpenBSD. OpenSSH is a little harder to build, but is preferable because it has a strong base of ongoing support, and is capable of communicating with commercial products using the SSH 2.0 protocol and therefore more flexible for your users. As before, you will need to build from the source on a separate host which has compilers installed and then move the binaries and libraries over to this machine. OpenSSH requires Zlib (which you should have added as a package above and which obviously needs to be installed on the compiling machine) and OpenSSL, which you will build here first. From the other machine:

1. ___ Download *ftp://ftp.openssl.org/source/openssl-0.9.6.tar.gz*
2. ___ Unpack: **gzip -dc openssl-0.9.6.tar.gz | tar xvf -**
3. ___ **cd openssl-0.9.6**
4. ___ **./config**
5. ___ **make && make test**
6. ___ **make install** (You need the libraries in the right place to compile OpenSSH.)
7. ___ Download *ftp://ftp.openbsd.org/pub/OpenBSD/OpenSSH/portable/openssh-2.3.0p1.tar.gz*
8. ___ Unpack: **gzip -dc openssh-2.3.0p1.tar.gz | tar xvf -**
9. ___ **cd openssh-2.3.0p1**
10. ___ **./configure --with-ssl-dir=/usr/local/ssl --sysconfdir=/etc/ssh --with-tcp-wrappers --with-ipv4-default**
11. ___ **make**
12. ___ **make install**

With both OpenSSL and OpenSSH built, you can transfer the files to the target host:

1. ___ On the compiler host, save */usr/local/lib/libssl.a*, */usr/local/lib/libcrypto.a*, the entire */usr/local/include/openssl* directory, the entire */etc/ssh* directory, */usr/local/bin/ssh** and */usr/local/sbin/ssh** onto removable media which is readable by

the target machine.

2. ___ Mount the removable media on the target machine.

3. ___ On the target host, copy all of the above files into the same locations where they were on the compiler host.

Configuring TCP Wrappers

Your users may connect from anywhere, but you must ensure that they only use SSH. You should also be notified when connection attempts are made and rejected. Let ADMIN_EMAIL be the appropriate email address for such messages to go on your site.

1. ___ **/bin/touch /etc/hosts.allow**

2. ___ **/bin/chown root:root /etc/hosts.allow**

3. ___ **/bin/chmod 600 /etc/hosts.allow**

4. ___ **/bin/echo 'ssh: ALL' > /etc/hosts.allow**

5. ___ **/bin/touch /etc/hosts.deny**

6. ___ **/bin/chown root:root /etc/hosts.deny**

7. ___ **/bin/chmod 600 /etc/hosts.deny**

8. ___ **/bin/echo 'ALL: ALL: /usr/bin/mailx -s "%s: connection attempt from %a" <ADMIN_EMAIL>' > /etc/hosts.allow**

Configuring SSH

1. ___ Modify */etc/ssh/sshd_config* to look like this:

```
Port 22
ListenAddress 0.0.0.0
SyslogFacility AUTH
LogLevel INFO

HostKey /etc/ssh_host_key
ServerKeyBits 1024
KeyRegenerationInterval 900

CheckMail no
UseLogin no
PrintMotd no
KeepAlive no

PermitRootLogin no
IgnoreRhosts yes
RhostsAuthentication no
RhostsRSAAuthentication no
RSAAuthentication yes
PasswordAuthentication yes
```

```
PermitEmptyPasswords no
StrictModes yes
UseLogin no
LoginGraceTime 180
```

2. ___ **/bin/chown root:root /etc/ssh/sshd_config**

3. ___ **/bin/chmod 600 /etc/ssh/sshd_config**

4. ___ Create a startup script */etc/init.d/sshd* which looks like this:

```
#!/bin/sh -
#
#
PIDFILE="/etc/sshd.pid"
SSHD=/opt/slocal/sbin/sshd

case $1 in
  start)
test -f $SSHD || exit 0
$SSHD
;;
  stop)
test -f $PIDFILE || exit 0
PID=`cat $PIDFILE`
test "$PID" && kill "$PID"
> $PIDFILE
;;
   *)
echo "Usage /etc/init.d/sshd {start | stop)";;
esac
exit 0
```

5. ___ **/bin/chown root:sys /etc/init.d/sshd**

6. ___ **/bin/chmod 744 /etc/init.d/sshd**

7. ___ **/bin/ln /etc/init.d/sshd /etc/rc2.d/S75sshd**

8. ___ Start SSHD: **/etc/init.d/sshd start** (Note: this will generate the server key.)

Configuring NTP

You installed the Solaris NTP package earlier, which created the startup scripts and stored the binaries and configuration file. But you still need to modify that file.

1. ___ Go to the list of public NTP servers at *http://www.eeics.udel.edu/~mills/ntp/servers.htm* and find three secondary servers and their administrative contact infomation.

2. ___ Contact the administrator for each of these three sites and ask for permission to connect.

3. ___ Once you have permission for three servers, add each server's IP address to

/etc/ntp.conf in the form: **server   IPADDRESS**

Further Tightening

Make the system stack non-executable:

1. **___** Append to */etc/system*: **set noexec_user_stack = 1**.
2. **___** Append to */etc/system*: **set noexec_user_stack_log = 1**.

Password aging is annoying, but on a machine this powerful, a brute force attack on a weak encryption algorithm like DES is possible given enough time. Against this threat, it's not unreasonable to expect users to change their password every three months.

1. **___** Set **MAXWEEKS=13** in */etc/default/passwd*.

Deny the ability to store core files. This makes debugging a little harder, but it prevents attackers from generating cores to create a DoS and from capturing password information in the core.

1. **___** Append to */etc/system*: **set sys:coredumpsize = 0**

Root should not be able to login directly, either remotely or at the console. This forces any legitimate attempt at a root shell to be via **su** and therefore logged. Actions can be traced back to the original user, or if the original user cannot be found in the log, you will know the action is malicious.

1. **___** Change "**CONSOLE=/dev/console**" to "**CONSOLE=**" in */etc/default/login*. (Note: the former prevents remote logins as root, the latter will deny root the ability to log in even at the console.)

Set a fairly restrictive umask for all users:

1. **___** Uncomment **UMASK=022** in */etc/default/login*.
2. **___** Add **umask 022** to */etc/profile* and */etc/.login*.

Create an account for yourself, so that you can login at the next reboot:

1. **___** **/bin/touch /home/<username>**
2. **___** **/usr/sbin/groupadd -g <gid> <group>**
3. **___** **/usr/sbin/useradd -u <uid> -g <gid> -d <home dir> -s <default shell> -c <full name> <loginid>**
4. **___** **/bin/passwd <loginid>**

TCP_STRONG_ISS sets the TCP initial sequence number generation parameters. Setting the value to 2 enables RFC 1948 sequence number generation, unique per connection ID. This makes it more difficult to hijack a session by predicting TCP sequencing.

1. **___** Set **TCP_STRONG_ISS=2** in */etc/default/inetinit*.

Also, the message in */etc/motd* and */etc/issue* needs to warn users that unauthorized activity is prohibited, that access may be logged, that users on the system consent to logging, and possibly that logs might be turned over to law enforcement if criminal activity is found. The wording and the rules themselves must be exact and carefully chosen. You will need to warn those who access your system, but that warning must be precise. Get it approved by your legal counsel and policy makers before you apply it.

1. ___ */etc/motd* and */etc/issue* approved by legal.
2. ___ */etc/motd* and */etc/issue* installed on the system.

Physical Security

If a machine is not physically secure, it is not secure. Anyone who has access to the hardware can bring down the system or manipulate the disk.

1. ___ The system should be installed in a data center environment. SPARCengine UltraAXmp's can run hot if not kept in a cool room, and UltraII CPUs are notorious for disliking humidity and dust.
2. ___ The data center should not be accessable to unauthorized personnel.
3. ___ The data center should have cameras installed to monitor and record all actions inside.
4. The persons who are authorized to access this machine must be trained and trusted. Have a supervisor sign here to testify that he or she has checked the qualifications and backgrounds of all authorized persons:

   Printed Name:____

   Signature:____ Date:________________
   _

Testing

Bring the machine online and test connectivity and for the desired results from the restrictions you imposed. If any test fails, bring the machine offline immediately and repair.

1. ___ Cannot boot from CD-ROM without *obp_password*.
2. ___ Cannot write to */usr* or */opt*.
3. ___ Cannot execute set-uid scripts from */opt*, */var* or */*.
4. ___ Cannot execute **sys-unconfig**.
5. ___ Cannot mount NFS volumes.
6. ___ Cannot export NFS volumes.
7. ___ Can SSH out to another machine.

8. ___ Can SSH into this machine.
9. ___ Ensure that you cannot telnet/rlogin/rsh/ftp into this machine.
10. ___ Cannot login as root via SSH.
11. ___ Cannot login as root to the console.
12. ___ */var/log/authlog* and */var/log/loginlog* are recording these attempts.
13. ___ Correct */etc/motd* appears at login.
14. ___ RPC processes are not running.
15. ___ NFS/autofs/cachefs are not running.
16. ___ Run **nmap** and **nessus** against the new host.
17. ___ Send email out and have it properly arrive.
18. ___ Send email in and have it properly received.
19. ___ System accounting logs have data.
20. ___ Audit logs have data.

Backup

The system should now be fairly secure. With the OS disk in this wonderful pristine state, back it up.

1. ___ Ensure that the host sees both disks. You can check this by running **format** and looking for both device paths. If the second disk does not appear, halt the system, check that the disk is attached and visable on the SCSI chain (run a **probe-scsi-all** at the **ok** prompt), and then reboot with **boot -r**. Once the machine is up, check **format** again.

2. What is the OS disk device (e.g. c0t0d0)? _______________. To get the raw device name you should prepend a */dev/rdsk/* and append an *s2*. (In the Sun disk label, "s2" is the "whole-disk" slice. It's a nice feature in a case like this because it allows us to dump the contents of a disk without knowing the specific partition layout.) Given that, what is the raw device name for the whole-disk slice of the OS disk (e.g. */dev/rdsk/c0t0d0s2*)? _________________________. Substitute this below when you see "RAW_WHOLE_PATH".

3. What is the backup disk device (e.g. c0t1d0)? _______________. Following the same syntax as with the OS disk, what is the raw device name for the whole-disk slice of the backup disk (e.g. */dev/rdsk/c0t1d0s2*)? _________________________. Substitute this below when you see "BK_RAW_WHOLE_PATH".

4. ___ **/bin/dd if=RAW_WHOLE_PATH of=BK_RAW_WHOLE_PATH bs=4096**
This dump will likely take a few hours. While it's running, make a list of the filesystem partitions for the backup disk. (You will need these entries for **fsck** below.) Above you

specified which OS disk slice was used for which filesystem. For instance, if slice "s6" was used for */usr* on the OS disk, it will also hold the */usr* filesystem on the backup disk. List the slice for each filesystem, and from it construct the raw path for the same slice on the backup disk. From our example, */usr* on the backup disk would be */dev/rdsk/c0t1d0s5*. Substitute these values for the variables when they are listed below.

5. Slice for / ______ BK_RAW_ROOT_PATH ______________________
6. Slice for */dev* ______ BK_RAW_DEV_PATH ______________________
7. Slice for */devices* ______ BK_RAW_DEVICES_PATH ______________________
8. Slice for */opt* ______ BK_RAW_OPT_PATH ______________________
9. Slice for */usr* ______ BK_RAW_USR_PATH ______________________
10. Slice for */var* ______ BK_RAW_VAR_PATH ______________________
11. Once the dump is complete, you should **fsck** the backup disk's filesystems to get them into a usable state. VERY IMPORTANT: Be sure that you **fsck** only the filesystems on the backup disk. If you made a mistake above and accidentally type in a path for a filesystem on the OS disk, you could damage that filesystem. So double-check your work above: ___
12. ___ **/etc/fsck -y BK_RAW_ROOT_PATH**
    Repeat until no messages appear saying the filesystem was modified.
13. ___ **/etc/fsck -y BK_RAW_DEV_PATH**
    Repeat until no messages appear saying the filesystem was modified.
14. ___ **/etc/fsck -y BK_RAW_DEVICES_PATH**
    Repeat until no messages appear saying the filesystem was modified.
15. ___ **/etc/fsck -y BK_RAW_OPT_PATH**
    Repeat until no messages appear saying the filesystem was modified.
16. ___ **/etc/fsck -y BK_RAW_USR_PATH**
    Repeat until no messages appear saying the filesystem was modified.
17. ___ **/etc/fsck -y BK_RAW_VAR_PATH**
    Repeat until no messages appear saying the filesystem was modified.
18. ___ When this is complete, remove the disk from the system.
19. ___ This is the sole backup to your secure system. You should make another backup on a different machine (ideally to optical read-only media, if available) and store this one in a very secure location.
20. ___ Run tripwire on the second backup. Keep a catalog of the checksums for future reference.

Remember that you have to repeat this section every time you make changes to the OS disk (patches or upgrades).

Your John Hancock

Well done. The host should be reasonably secure and prepared to serve its users. Obviously a great deal of attention should be paid daily. Logs need to be watched carefully, user behavior should be monitored, and basic system administration will still take place. But hopefully following the instructions above made all of those tasks a little easier. Sign below to certify that the system is ready for prime time.

Printed Name:_______________________________________

Signature:_______________________________________ Date:_______________

Last update: 22-Nov-2000