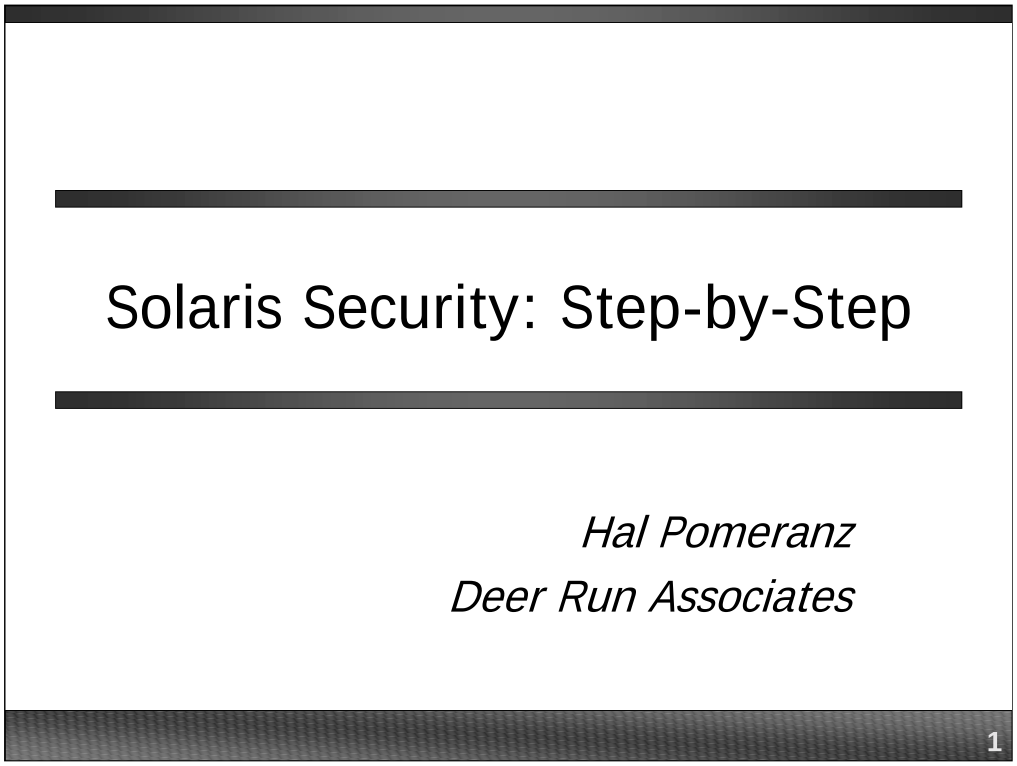


Solaris Security: Step-by-Step

Hal Pomeranz

Deer Run Associates

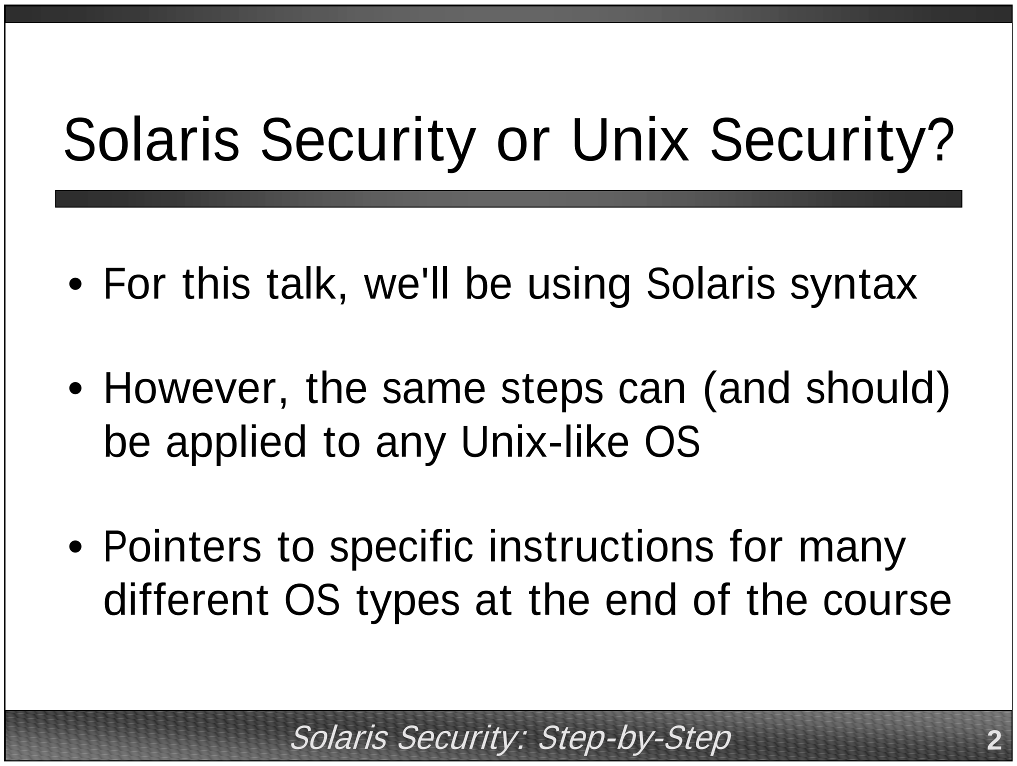
1

Hal Pomeranz * Founder/CEO * *hal@deer-run.com*

Deer Run Associates * PO Box 20370 * Oakland, CA 94620-0370

+1 510-339-7740 (voice) * +1 510-339-3941 (fax)

http://www.deer-run.com/

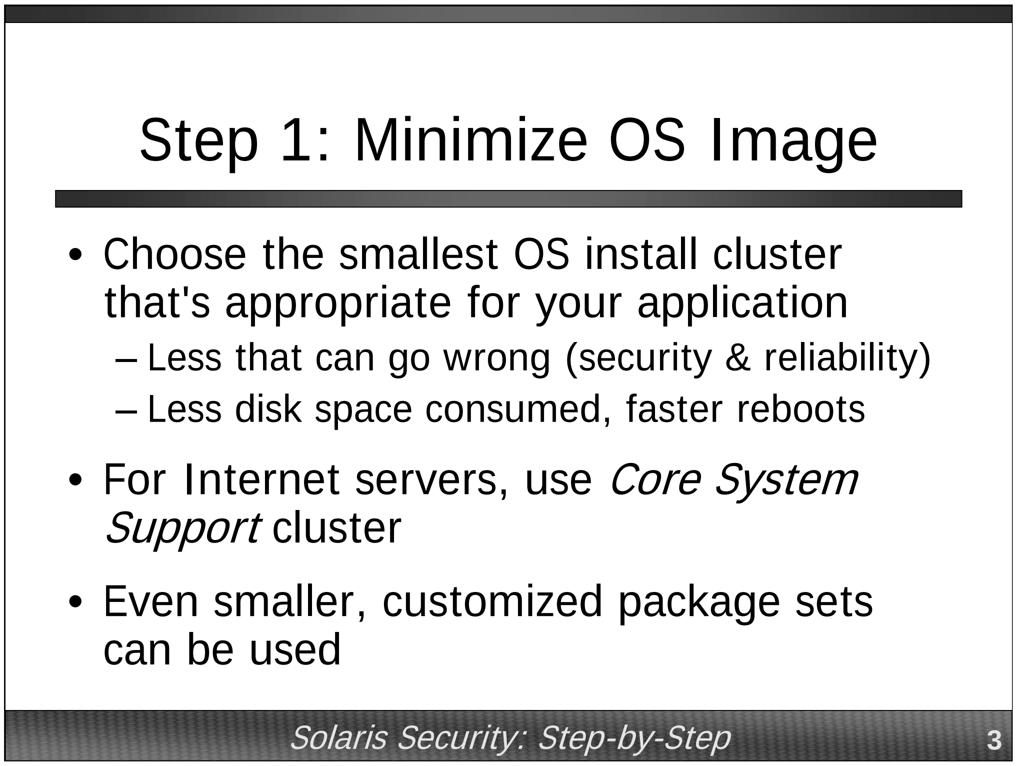


Solaris Security or Unix Security?

- For this talk, we'll be using Solaris syntax
- However, the same steps can (and should) be applied to any Unix-like OS
- Pointers to specific instructions for many different OS types at the end of the course

Solaris Security: Step-by-Step 2

While this talk will be looking primarily at the Solaris operating system, the 10 basic steps we'll be covering can be applied to any Unix-like operating system. The trick is finding out the correct syntax for each vendor's operating system. At the end of the talk, there are two pages of URLs which point to various Internet sites that either have automated tools for hardening systems, or white papers and other documentation on system hardening.



Step 1: Minimize OS Image

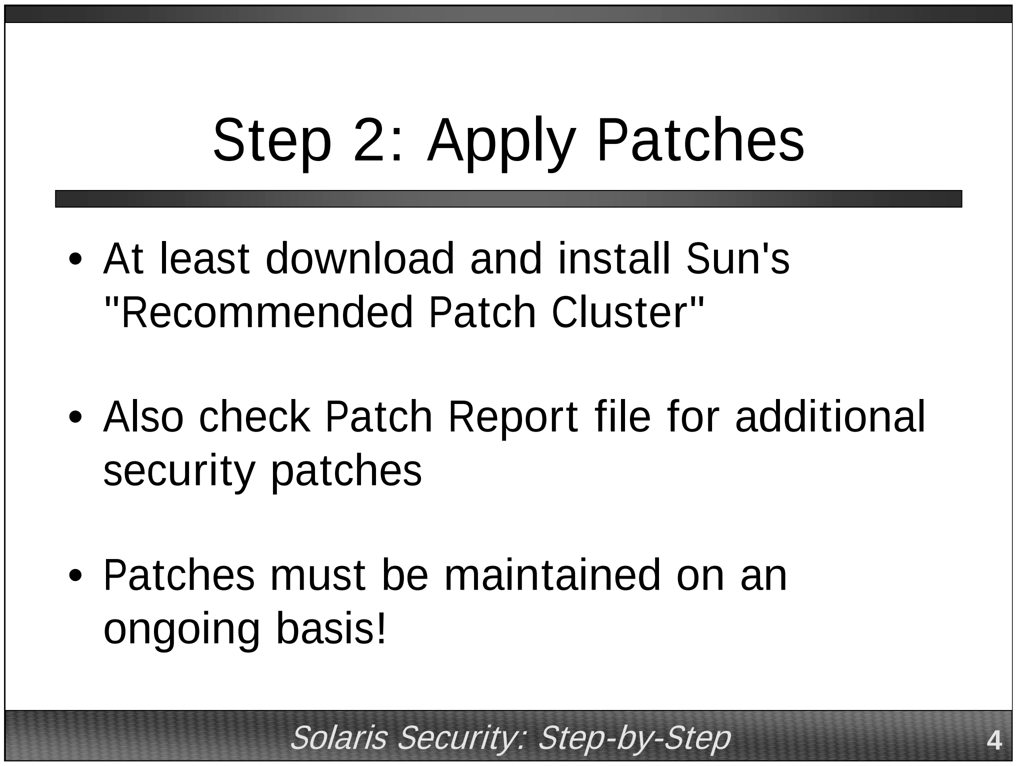
- Choose the smallest OS install cluster that's appropriate for your application
  - Less that can go wrong (security & reliability)
  - Less disk space consumed, faster reboots
- For Internet servers, use *Core System Support* cluster
- Even smaller, customized package sets can be used

Solaris Security: Step-by-Step 3

The first step is to pick the smallest operating system image you can get away with for your application. Extra features like Windowing systems and GUI-based management tools, volume managers, etc. are convenient but can provide extra avenues for attackers to break into your systems. We want to tend towards the "security" end of the "security vs. ease-of-use" spectrum.

Internet servers– like Web servers, FTP servers, firewall devices, etc.– really only need a bare minimum OS install. For Solaris, this is the *Core System Support* cluster (`SUNWCreq`). This image does not include the Windowing system, programming tools (`/usr/ccs/bin`, header files in `/usr/include`, etc.), or even the system manual pages. You may want to customize the package list further for your particular application. Sun has published a white paper on further minimizing the Solaris operating system as part of their Blueprints™ series (URL at the end of the talk).

One of the advantages to this kind of stripped-down OS install is size. The full Solaris install requires at least a gigabyte of space just for the OS files themselves (not counting swap space, space for logging, user and application data, etc). The *Core* cluster can fit in a couple of hundred megabytes. Also, with less installed on the system, there's less that can go wrong to cause the system to crash or lock up, and rebooting or restarting the system happens much faster.



Step 2: Apply Patches

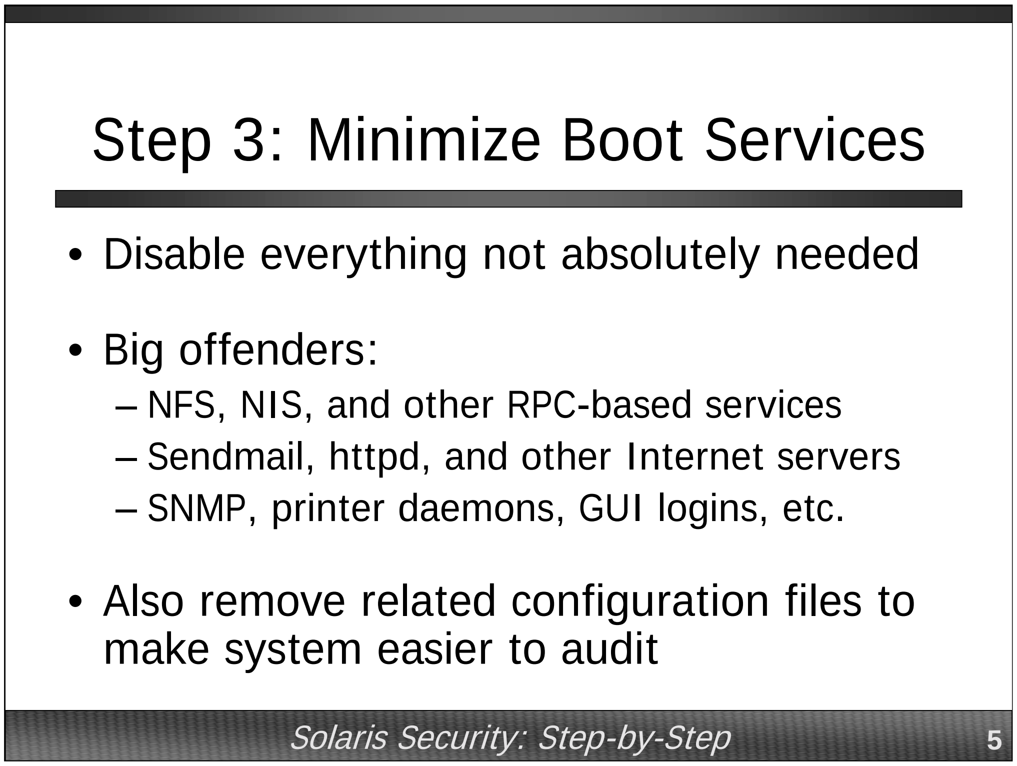
- At least download and install Sun's "Recommended Patch Cluster"
- Also check Patch Report file for additional security patches
- Patches must be maintained on an ongoing basis!

Solaris Security: Step-by-Step 4

Once you've decided exactly which pieces of the operating system you wish to install, download and install the Recommended Patch Cluster for your OS version. It's important that you install all of the OS packages that you'll need *before* you apply patches. If you install OS software after your patch install, you may end up with unpatched software that has security issues.

Patches can be found at `ftp://sunsolve.sun.com/pub/patches`: the cluster files are named <*vers*>`_Recommended.zip` (or `.tar.Z` for Solaris 2.6 and earlier). Note that not all security patches are necessarily included in the recommended patch set, so you'll also want to check out the `Solaris`<*vers*>`.PatchReport` files in the same directory.

New patches are coming out all the time, so figure out some mechanism for keeping your machines up-to-date. If you're running a big farm of identical Internet servers, you may just want to rebuild your machines in rotation (a few at a time) and install up-to-date patches at that point.



Step 3: Minimize Boot Services

- Disable everything not absolutely needed
- Big offenders:
  - NFS, NIS, and other RPC-based services
  - Sendmail, httpd, and other Internet servers
  - SNMP, printer daemons, GUI logins, etc.
- Also remove related configuration files to make system easier to audit

Solaris Security: Step-by-Step 5

Patches cover security vulnerabilities that we're aware of, but new exploits are being discovered every day. The best way to protect yourself from problems that we don't know about yet is to turn off all of the services that you're not using. There's an unbeatable smug feeling of satisfaction you get reading the latest BUGTRAQ posting a knowing that you're not vulnerable because you elected not to run a given service in the first place.

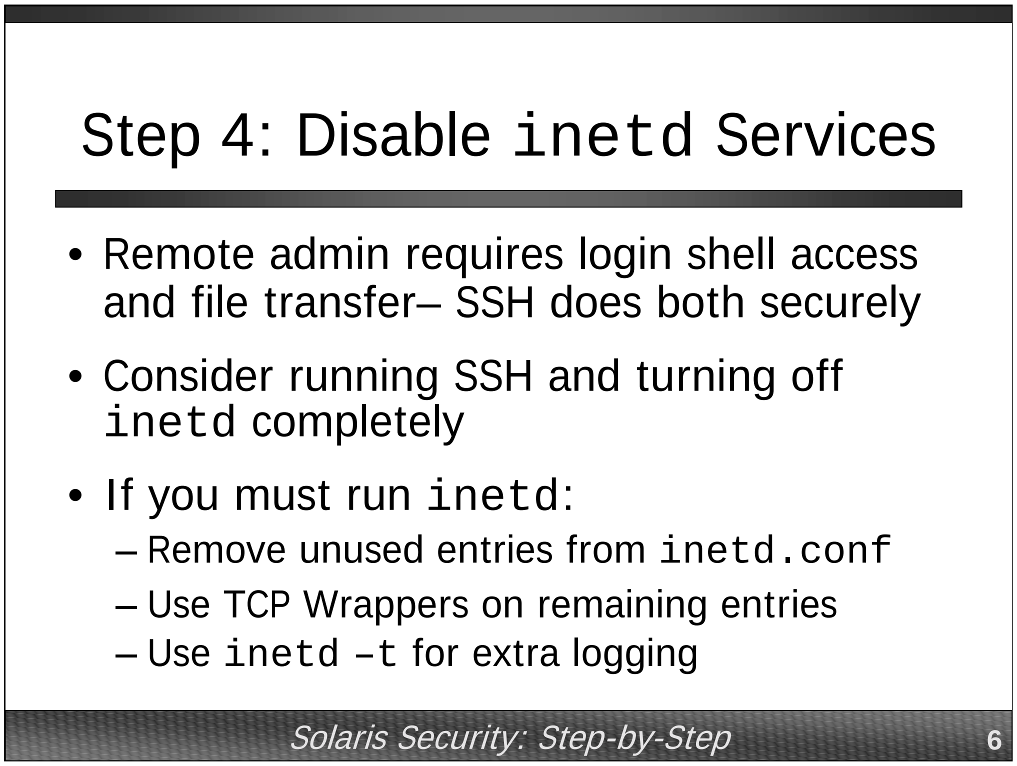
The basic principal here is:

If you don't need it, turn it off.

If you're not sure whether you need it or not, turn it off and see what breaks!

Prior to the distributed denial-of-service attacks in early Y2K, thousands of Internet-connected machines were compromised via the well-known `rpc.ttdbserverd` and `rpc.cmsd` holes. These machines later became the "zombie" or "daemon" servers used in the attacks. You have to wonder why Web servers need to be running Sun's calendar manager daemon– the answer being, of course, that they don't!

Certain utilities are fine inside of a strongly firewalled environment– NFS, NIS, print daemons, GUIs, etc.– but should never be used on machines that are essentially connected directly to the Internet (Web servers, mail relays, etc.). Given the rash of SNMP vulnerabilities in the last few months, definitely disable the Solaris SNMP daemon if you're not currently using SNMP for network management. Also, unless your system is a mail server, turn off the Sendmail daemon or at least don't listen for incoming mail on port 25 (disable the `-bd` switch).



Step 4: Disable `inetd` Services

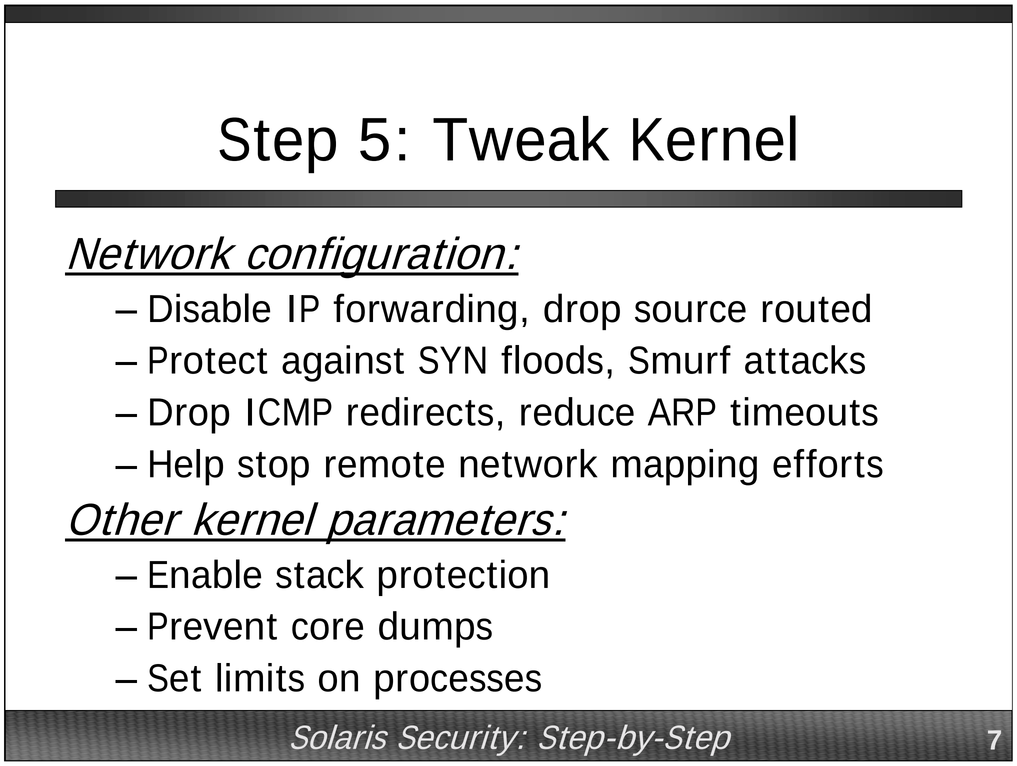
- Remote admin requires login shell access and file transfer– SSH does both securely
- Consider running SSH and turning off `inetd` completely
- If you must run `inetd`:
  - Remove unused entries from `inetd.conf`
  - Use TCP Wrappers on remaining entries
  - Use `inetd –t` for extra logging

Solaris Security: Step-by-Step 6

In addition to the other services started at boot time, `inetd` will start up a number of other network-related services on demand. Everything that is run out of `inetd` has probably had at least one security vulnerability reported against it in recent memory. `inetd` enables clear-text login and file transfer protocols (like `telnet`, FTP, and `rlogin`/`rcp`) which can be sniffed, spoofed, and hijacked. Other services (like `echo`, `chargen`, etc.) can be used as denial-of-service attacks.

From a networking perspective, all you really need is the ability to administer your systems remotely (and perhaps not even that if you're willing to do all your work at the system console). This means you need the ability to log into the system over the network and transfer files back and forth. SSH provides both of these services (and more) and is encrypted to prevent eavesdropping and hijacking. It may be that all you need is SSH– in which case you can turn off `inetd` completely.

If you must run `inetd` for some reason, make sure to eliminate all services that are not absolutely required and use TCP Wrappers to protect the rest. Solaris also supports "connection tracing" in `inetd` (start `inetd` with the `–t` flag) which provides additional logging about each connection.



Step 5: Tweak Kernel

Network configuration:

- Disable IP forwarding, drop source routed
- Protect against SYN floods, Smurf attacks
- Drop ICMP redirects, reduce ARP timeouts
- Help stop remote network mapping efforts

Other kernel parameters:

- Enable stack protection
- Prevent core dumps
- Set limits on processes

Solaris Security: Step-by-Step 7

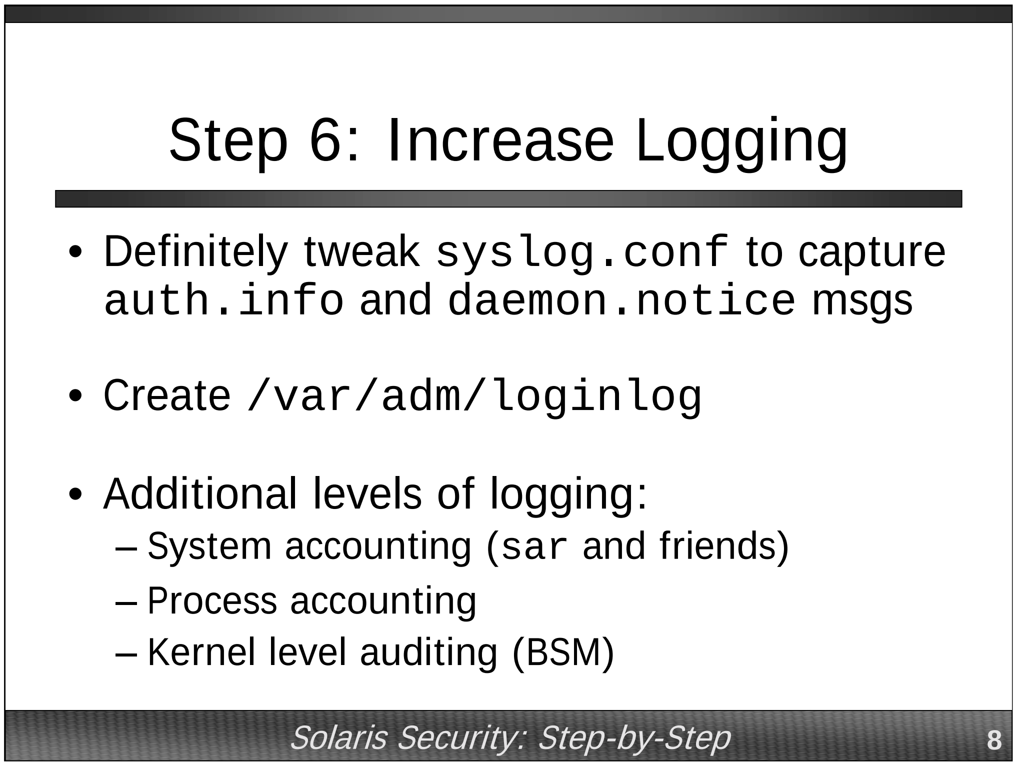
There are a number of parameters in the Solaris kernel which can be tweaked to enhance security. Network parameters are generally set using the `ndd` command (you'll need to add a script to your boot directories which sets these parameters automatically), and other parameters can be set in `/etc/system`.

On the network side of things, Solaris systems by default have IP forwarding enabled (the system will act as a router if it has multiple network interfaces) and source routed packets will be accepted. Neither of these is a good idea. The default Solaris ARP timeout (20 minutes) makes ARP spoofing attacks much easier, so tuning this value down can help. Increasing the values for number of half-open connections and reducing the half-open connection timeout can help your system handle SYN flood type attacks more easily. Disabling various types of ICMP messages can help you prevent attackers from mapping your networks remotely, and even prevent your systems from being used as an amplifier network for a Smurf-style attack. Similarly, you can prevent the machine from obeying ICMP redirects (which could be used to maliciously change your routing table on the fly).

You should definitely add the following two lines to your `/etc/system` file:

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

This turns on "stack protection" (available for Solaris 2.6 and later only) which will help protect you from many buffer overflow attacks. You may also want to disable core dumps in `/etc/system` on your production servers (core dumps are world-readable and can contain sensitive information), but remember that your developers will probably want the ability to get core files on their development workstations. `/etc/system` can also allow you to set other limits (like the maximum number of processes per user, etc.) that can help prevent local denial-of-service attacks.



Step 6: Increase Logging

- Definitely tweak `syslog.conf` to capture `auth.info` and `daemon.notice` msgs
- Create `/var/adm/loginlog`
- Additional levels of logging:
  - System accounting (`sar` and friends)
  - Process accounting
  - Kernel level auditing (BSM)

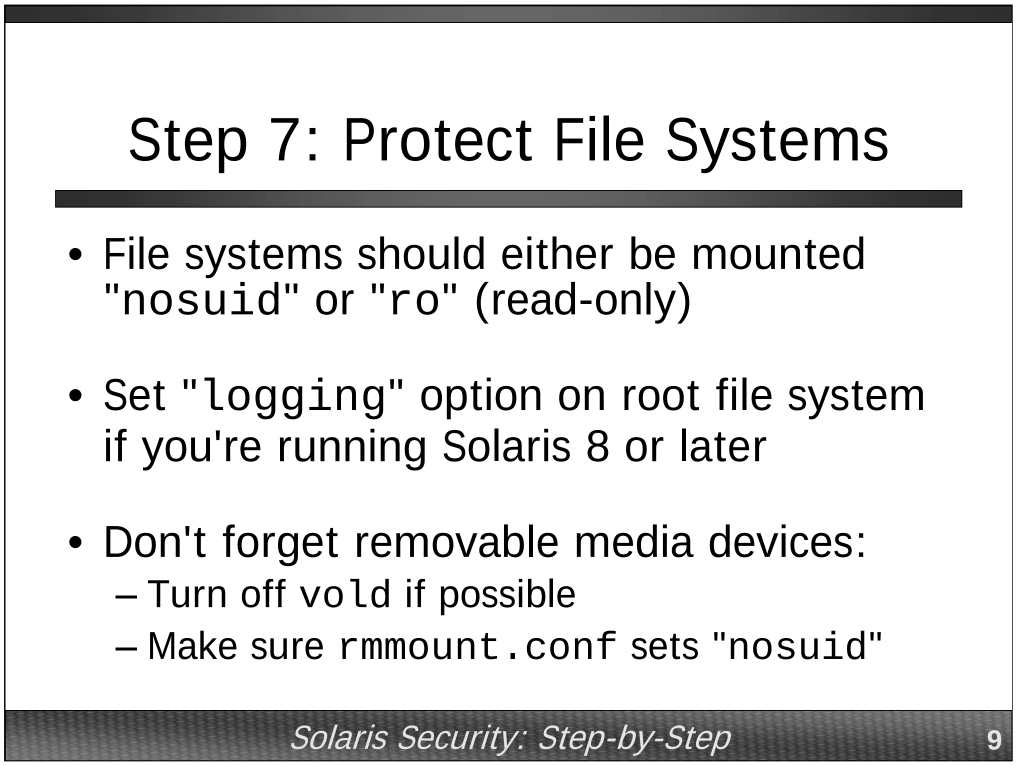
Solaris Security: Step-by-Step 8

The more information you log about your systems, the more likely you are to log something which enables you to detect an attacker. Think about Cliff Stoll noticing an intruder on his system because of a 75 cent accounting discrepancy (read The Cuckoo's Egg for more information).

At a minimum, tweak your `syslog.conf` file so that you at least log `auth.info` (and higher) to a local log file and/or to some other machine. By default Solaris throws messages sent to `LOG_AUTH` away, which is unfortunate since this is where all of the interesting security information about the system goes. If you're using `inetd` connection tracing (`inetd -t`), then you also need to log `daemon.notice` in order to get the connection logs from `inetd`.

If you create `/var/adm/loginlog`, then bad login messages will be logged to this file (and to `LOG_AUTH`). Starting with Solaris 8, you can also set `SYSLOG_FAILED_LOGINS` in `/etc/default/login` to control how many failed logins must occur before a message is logged. If set to zero, then all failed logins are logged (hint: this is what you want).

There are several other options beyond these sorts of standard logging facilities. System accounting keeps track of system usage information (CPU load, memory usage, disk usage, etc.)– by defining a baseline for your system's performance, you can detect unexpected or illicit usage by detecting departures from the baseline. Process accounting keeps track of which commands users are running on your system, but can be a performance drag on the machine. Kernel level auditing (aka BSM) keeps track of even more information, but generates huge audit logs and can also be a performance drag.



Step 7: Protect File Systems

- File systems should either be mounted "`nosuid`" or "`ro`" (read-only)
- Set "`logging`" option on root file system if you're running Solaris 8 or later
- Don't forget removable media devices:
  - Turn off `vold` if possible
  - Make sure `rmmount.conf` sets "`nosuid`"

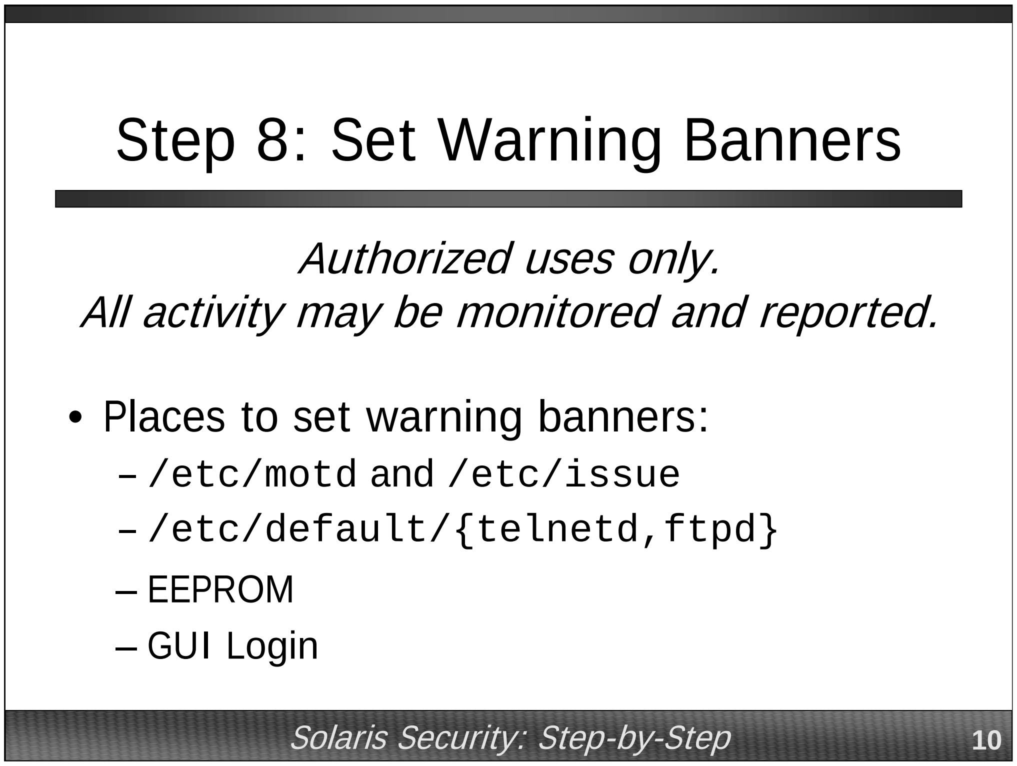
From a file system perspective, you want to make it harder for attackers to replace OS programs with their rootkits and other tools. You also want to prevent people from bringing unauthorized set-UID binaries on your system. The simplest bit of wisdom here is the basic rule:

File systems should either be mounted "`nosuid`" or "`ro`".

In particular, the `/usr` file system contains all of the critical OS programs and set-UID binaries but is relatively static, so you should mount `/usr` "`ro`" (read-only). When you need to install patches, you can make the file system read-write again with the command "`mount –o remount,rw /usr`", but you'll need to reboot the system to make it read-only again. You may be able to mount `/opt`, `/usr/local`, and other file systems containing third-party software tools "`ro`" as well.

All other file systems should be mounted "`nosuid`". Unfortunately, Sun has decided that setting "`nosuid`" also implies "`nodev`" (device files don't work), which means you can't mount the root file system (which contains `/devices`) "`nosuid`"– and you can't make it read-only either. You also need to be careful about the whole "`nosuid`"/"`nodev`" thing on file systems where you're running `chroot()`ed daemons (FTP servers, BIND, etc.) since the "`nosuid`" will interfere with the device files required in your `chroot()` directory structure.

Removable media is a great way for attackers to compromise your systems if they have physical access– just bring in a CD-ROM with a set-UID copy of the shell on it. If your users don't need the ability to mount CDs and floppies, turn off the system volume manager (you can use `sudo` to give your users this ability without having to run the volume manager). Also, be sure to set the "`nosuid`" option for all removable media devices in the volume manager configuration file, `/etc/rmmount.conf` (this is the default for Solaris 8 and later).



Step 8: Set Warning Banners

Authorized uses only.
All activity may be monitored and reported.

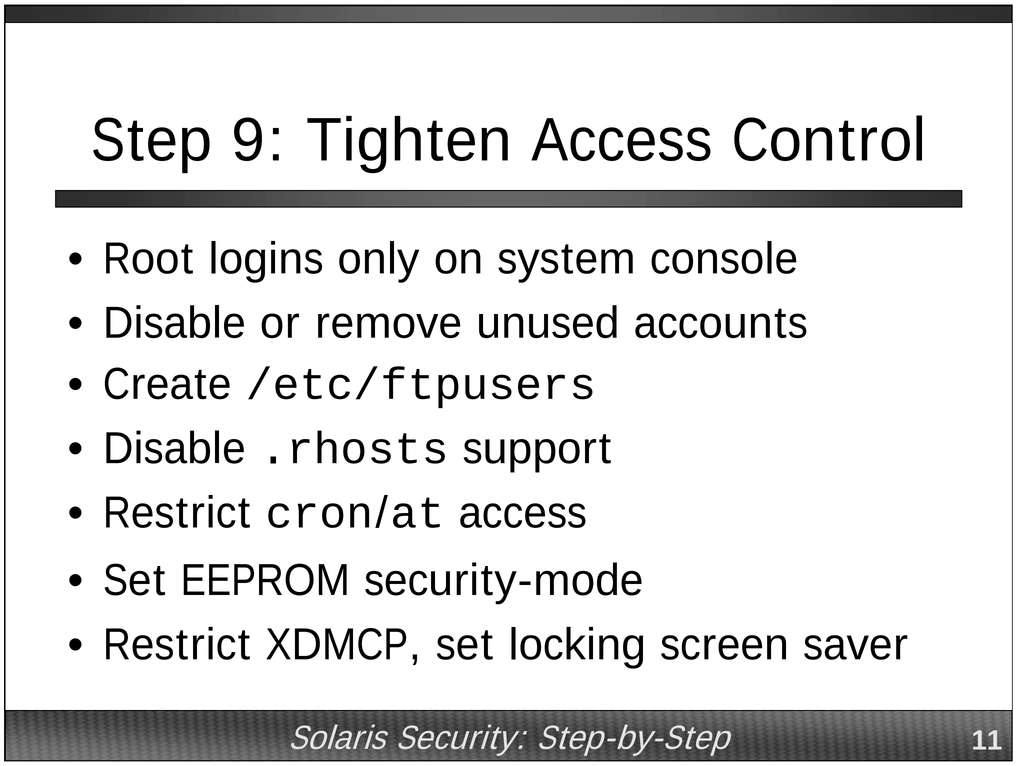
- Places to set warning banners:
  - `/etc/motd` and `/etc/issue`
  - `/etc/default/{telnetd,ftpd}`
  - EEPROM
  - GUI Login

Solaris Security: Step-by-Step 10

While the legal precedents for various system warning messages are unclear, security experts generally believe it's a good idea to inform users (authorized and otherwise) that the system is for authorized users/uses only and that all activity may be monitored and the results of that monitoring reported to the appropriate authorities as necessary. This message should definitely appear in the `/etc/motd` file (displayed *after* login). Ideally, though, you'd like to present the message before login. However, there are lots of different places you'll need to set the message because logins can occur over a variety of channels including the standard system login (put message in `/etc/issue`), `telnet` and FTP (use the `BANNER` variable in `/etc/default/{telnetd,ftpd}`), and GUI logins (`/etc/dt/config/*/Xresources`).

Here's a longer warning banner that was developed by the US Department of Justice:

> 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.



Step 9: Tighten Access Control

- Root logins only on system console
- Disable or remove unused accounts
- Create `/etc/ftpusers`
- Disable `.rhosts` support
- Restrict `cron/at` access
- Set EEPROM security-mode
- Restrict XDMCP, set locking screen saver

Solaris Security: Step-by-Step 11

• Make sure that `CONSOLE=/dev/console` is set in `/etc/default/login` so that root logins are only allowed on the system console device. Normally users should be forced to log in and then `su` to become `root`.

• Audit your password file and remove accounts that are not being used (like the `uucp`, `nuucp`, and `smtp` users). All other "system" accounts (like `adm`, `daemon`, `bin`, etc.) should have their shells set to `/dev/null` to block access.

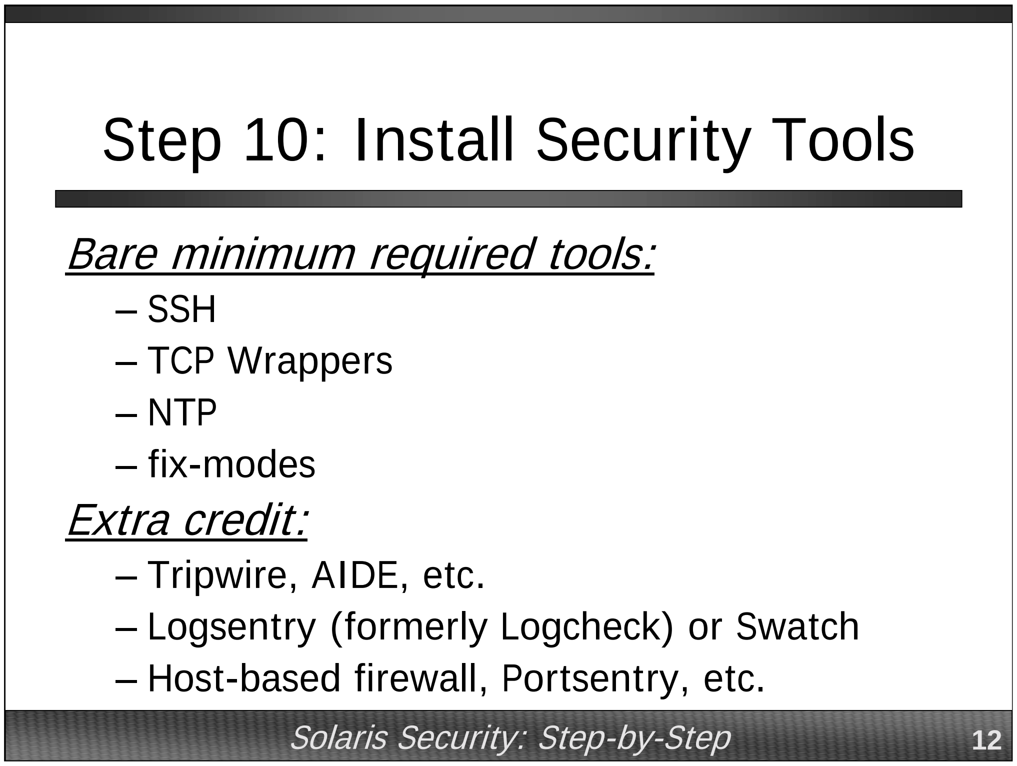
• Create an `/etc/ftpusers` file (even if you don't have FTP enabled). Remember that `/etc/ftpusers` is the list of users who *are not* allowed to FTP into the system– root belongs here, as do all of the "system" accounts from `/etc/password`.

• If you're using SSH, then make sure your `sshd_config` file does not allow `.rhosts` functionality ("`IgnoreRhosts yes`", etc.). Also edit `/etc/pam_conf` and remove any lines containing `rhosts_auth`, even if you've disabled `rlogin`/`rcp`.

• `cron.allow` and `at.allow` list the users who are allowed to run the `crontab` and `at` commands to submit/modify/delete `cron` jobs. On production systems, it is often the case that only the `root` user needs this functionality. Note that `cron` jobs will run as other users even if `root` is the only user listed in `cron.allow`.

• Setting "`eeprom security-mode=command`" will cause the machine to prompt for a password before boot-level commands are accepted. This prevents attackers with physical access from booting from alternate media (like a CD-ROM) and bypassing your system security.

• If you're running X Windows on the machine, make sure to disable remote XDMCP access in `/etc/dt/config/Xaccess`. You may also want to set a default locking screensaver timeout for your users in `/etc/dt/config/*/sys.resources`.



Step 10: Install Security Tools

Bare minimum required tools:

- SSH
- TCP Wrappers
- NTP
- fix-modes

Extra credit:

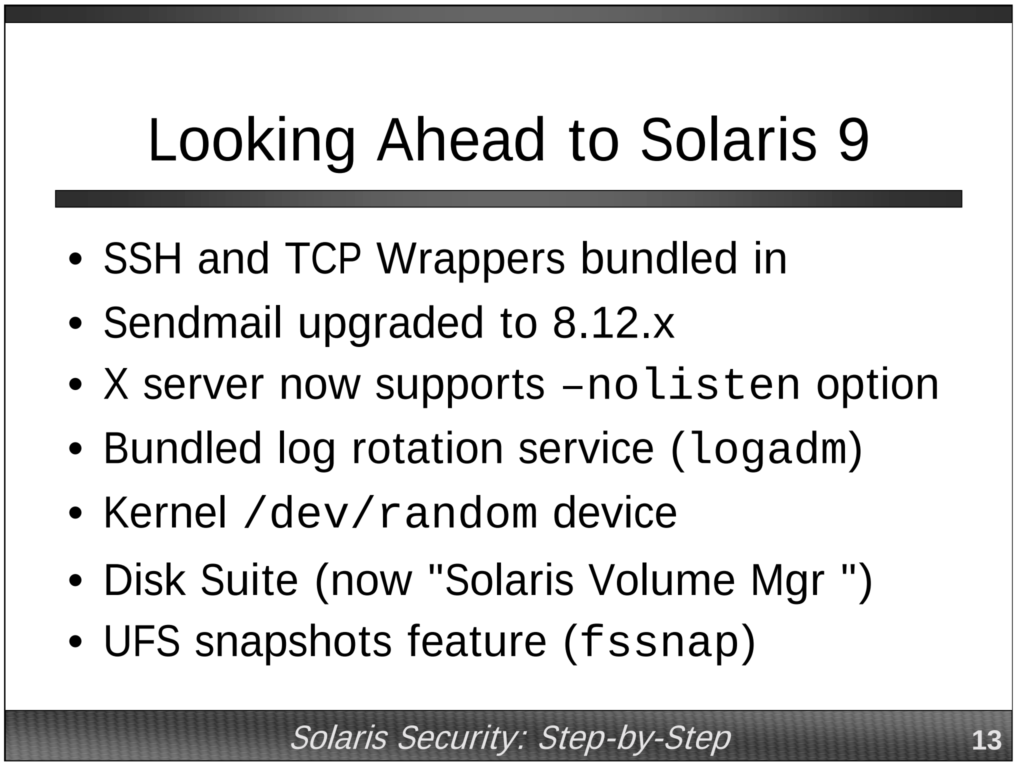
- Tripwire, AIDE, etc.
- Logsentry (formerly Logcheck) or Swatch
- Host-based firewall, Portsentry, etc.

Solaris Security: Step-by-Step 12

It's pretty much impossible to completely secure a Solaris machine without adding some freely-available security software from the Internet. `fix-modes` sets appropriate permissions on various OS files and directories (get `fix-modes` from `ftp://ftp.science.uva.nl/pub/solaris/`). SSH and TCP Wrappers are critical for network security. NTP (the Network Time Protocol for keeping system clocks in synch) doesn't necessarily seem like security software, but how are you planning on investigating security incidents at your site if the time on your hosts doesn't agree with the time on your routers, firewalls, and intrusion detection systems? SSH and TCP Wrappers are bundled with the OS starting with Solaris 9. NTP has been available in the Solaris OS since Solaris 2.6.

Beyond these tools, there are many other useful security tools out there. Tripwire, AIDE, et al are all integrity checking tools which will tell you when files on your machine have been modified (like after a successful break-in). Logsentry (formerly Logcheck) and Swatch are tools which will automatically monitor your log files and report "interesting" events to the administrator. You can also deploy host-based firewalls (IPFilter is the free tool for Solaris and other Unix systems, but Sun making their SunScreen product available for free as of Solaris 8) and monitoring tools like Portsentry.

`www.sunfreeware.com` provides pre-compiled binaries for many of these tools in Solaris `pkgadd` format.



Looking Ahead to Solaris 9

- SSH and TCP Wrappers bundled in
- Sendmail upgraded to 8.12.x
- X server now supports `–nolisten` option
- Bundled log rotation service (`logadm`)
- Kernel `/dev/random` device
- Disk Suite (now "Solaris Volume Mgr ")
- UFS snapshots feature (`fssnap`)

Solaris Security: Step-by-Step 13

Solaris 9 is now available and includes many, many useful new features (for more detail, check out `http://docs.sun.com/?p=/doc/806-5202/`):

• Solaris 9 now ships with SSH and TCP Wrappers (as well as many other "Open Source" tools like `bash`, `zsh`, GNU `grep`, GNU `tar`, etc).

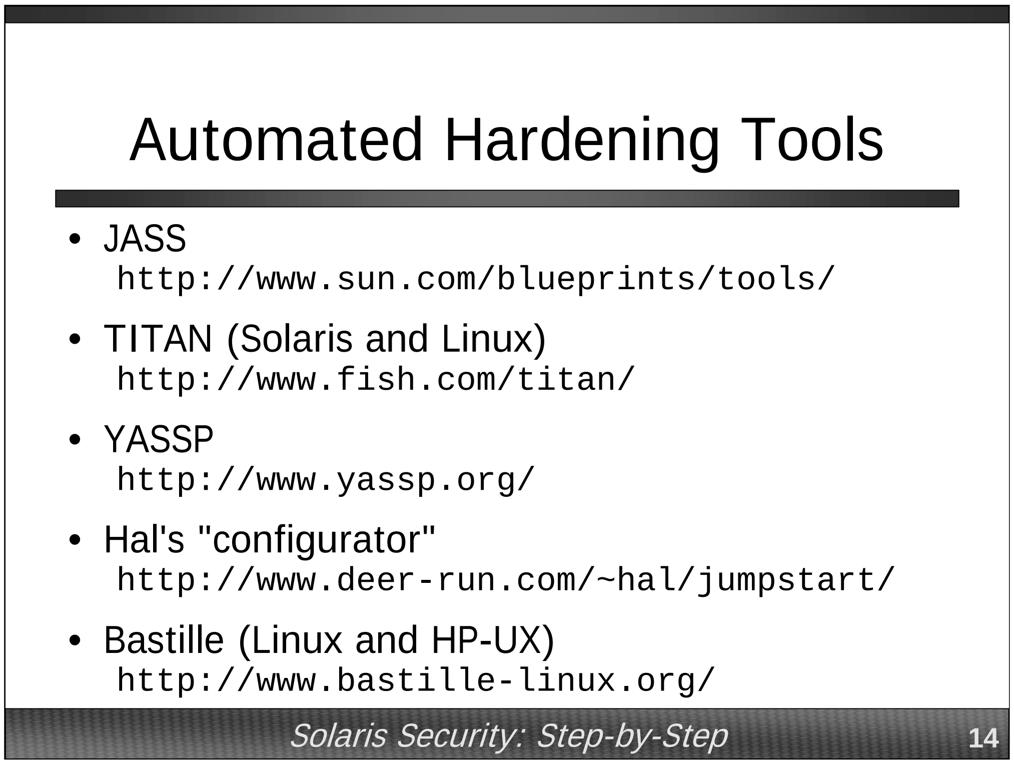
• Solaris 9 ships with Sendmail 8.12.x by default. This is significant because it means that `/usr/lib/sendmail` is no longer set-UID to `root`.

• The X server that ships with Solaris 9 now has a `–nolisten` option that enables the admin or user to prevent their X server for listening for X events over the network. Since local clients are still OK, users can still tunnel remote X events via SSH.

• Solaris 9 finally includes a generic log rotation/archiving tool called `logadm`. This means you don't have to worry about `/var/log/authlog` et al growing without bound and consuming `/var`.

• Solaris 9 finally includes a kernel `/dev/random` device for SSH, SSL, etc.

• Lot's of features have been added to the file system. Most interestingly, Solaris' UFS file system now supports "snapshots"– temporary, copy-on-write style, static read-only copies of the file system for easy backups, etc (Network Appliance users will be familiar with this concept). An updated version of Sun's Disk Suite product (now called the Solaris Volume Manager) is also bundled into the OS.



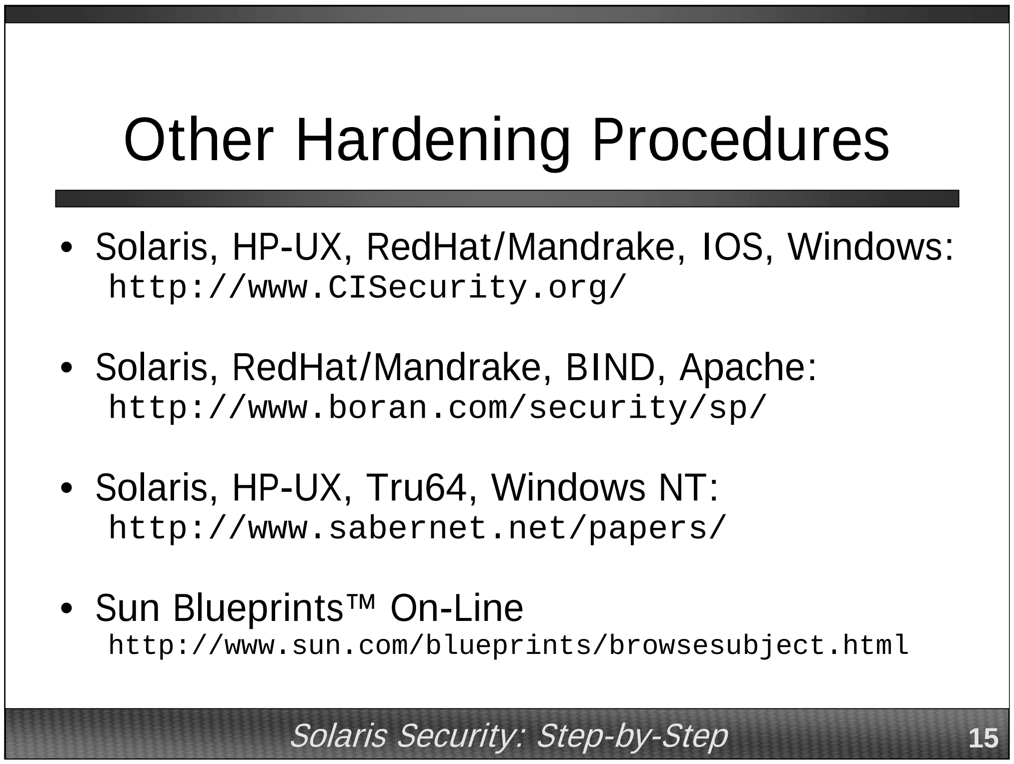
Automated Hardening Tools

- JASS
  `http://www.sun.com/blueprints/tools/`
- TITAN (Solaris and Linux)
  `http://www.fish.com/titan/`
- YASSP
  `http://www.yassp.org/`
- Hal's "configurator"
  `http://www.deer-run.com/~hal/jumpstart/`
- Bastille (Linux and HP-UX)
  `http://www.bastille-linux.org/`

Solaris Security: Step-by-Step 14

The first four tools listed here are all automated tools for hardening Solaris systems in similar ways to the material I've covered in this talk. Bastille does a similar job for Linux systems (and for HP-UX as of the most recent beta release). TITAN now supports both Solaris and Linux systems. YASSP development may be "dead in the water" because the author of the tool no longer has time to maintain it.

My own home page (`http://www.deer-run.com/~hal/`) has a bunch of other security-related information and useful tricks.



Other Hardening Procedures

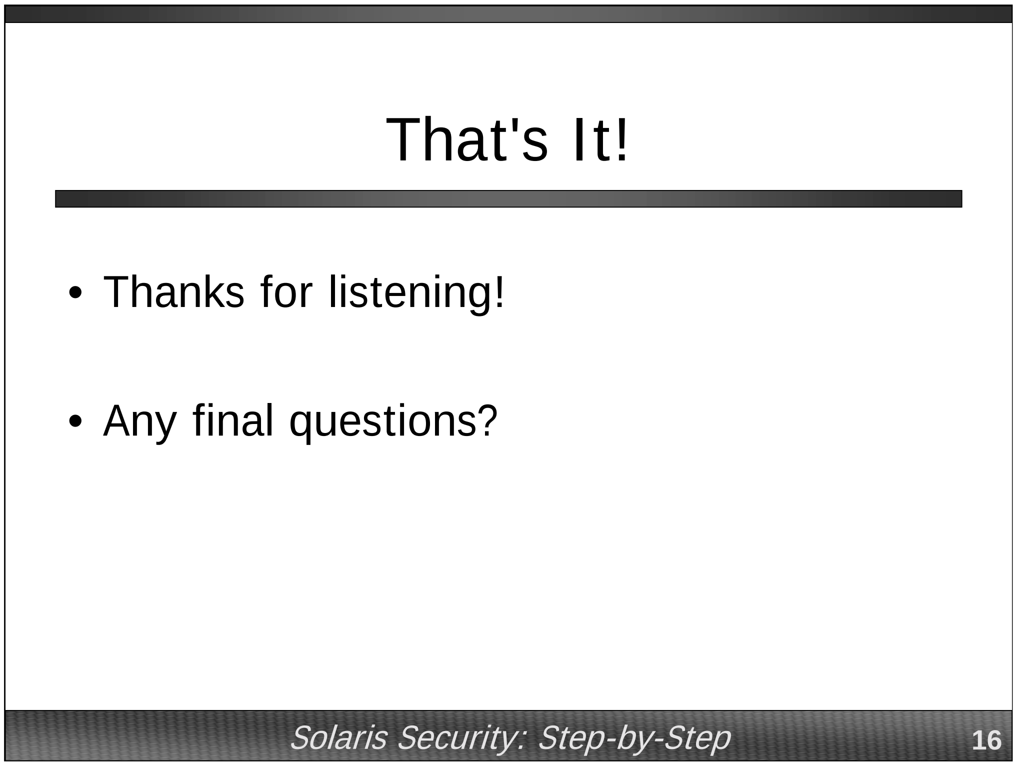
- Solaris, HP-UX, RedHat/Mandrake, IOS, Windows:
  `http://www.CISecurity.org/`
- Solaris, RedHat/Mandrake, BIND, Apache:
  `http://www.boran.com/security/sp/`
- Solaris, HP-UX, Tru64, Windows NT:
  `http://www.sabernet.net/papers/`
- Sun Blueprints™ On-Line
  `http://www.sun.com/blueprints/browsesubject.html`

Solaris Security: Step-by-Step 15

Here are some additional URLs which point to hardening guidelines for a variety of operating systems and applications.

In addition to free guidelines for various operating systems, the Center for Internet Security is also providing free auditing tools to test your systems for compliance. Unlike the automated "hardening" tools on the previous slide, the Center's tools are strictly "read-only" and don't change any settings on the machine.

Sun's Blueprints™ series has contains lots of interesting documents. In particular there are several documents by Alex Noordergraaf and others on topics ranging from OS minimization, to network security settings in the kernel, to the JASS toolkit.



That's It!

- Thanks for listening!
- Any final questions?

Solaris Security: Step-by-Step 16

This space intentionally left blank.