

Auditing Solaris Security with CLI

by Boris Loza, PhD, CISSP

Did you know that you can inspect the security on an existing Solaris box by using the command line interface (CLI)? You don't have to install any expensive GUI-based applications. In this article, we'll build a security check list using just native Solaris OS commands. Following this list step by step will help you to identify whether the system fits with your security policy.

In this article we won't go into specifics about the need for particular checks. For a detailed explanation about Solaris security, please refer to Practical UNIX and Internet Security by Simson Garfinkel and Gene Spafford. Note that the pound sign (#) in front of a UNIX command indicates that this command should be executed by root.

Gathering background information

The first step is to get information about your system. You do this with the following `uname` command:

```
uname -a
```

Using this output, we can tell the OS version (e.g., `SunOS 5.6`) the name of the hardware implementation (e.g., `sun4m sparc SUNW, Ultra-2`) and whether the latest kernel patch has been installed. Getting `Generic_105181-22` for Solaris 2.6 SPARC indicates that you have the latest kernel patch (at the time this article was written). You can obtain the latest kernel patch from <http://sunsolve.sun.com/>.

You need to know the OS version and the hardware implementation for applying OS/hardware-specific security patches. To display the patches installed, type the following:

```
showrev -p
```

This prints all patches currently installed on the system. You can compare the output with the list of recommended security patches available for this OS version. The latest recommended security patches are also available from SunSolve.

Checking account security

The next step is to check your user accounts. First, display accounts without a password:

```
#logins -p
```

Then, delete such accounts immediately or set passwords for them. Now, you check accounts with duplicate `UIDs`:

```
#logins -d
```

You'll want to provide a different UID for all accounts on your system.

Now you can display the date of the last password change, minimum number of days required between password changes, and maximum number of days the password is valid:

```
#passwd -sa
```

To alter any of these password attributes, edit the `/etc/default/passwd` file. Now, display inconsistencies in the password file:

```
/usr/sbin/pwck
```

This tells you about accounts with no login directory and the wrong shell. Next, display any inconsistencies in the group file:

```
/usr/sbin/grpck
```

Now you can check the accuracy of file attributes of installed files:

```
/usr/sbin/pkgchk -a
```

You can fix any inconsistencies you find manually. In addition, you may want to run the `fix-modes` utility found at <ftp://ftp.fwi.uva.nl/pub/solaris/fix-modes.tar.gz>. It will fix all mode 755 directories and binaries and change the ownership to root where needed. Currently it supports Solaris 2.2 to Solaris 8.

Now you can display system parameters with the following:

```
cat /etc/default/login
```

Pay attention to the `CONSOLE`, `PASSREQ` and `UMASK` variables. The `CONSOLE` and `PASSREQ` variables must be uncommented. `UMASK` should be set to 022 or 025.

Next, display the password, the shadow and the group files with the following:

```
cat /etc/passwd  
#cat /etc/shadow  
cat /etc/group
```

Make sure that `/dev/null` is the shell for all non-root users in `/etc/passwd`. See if you have `NP` no password entry for all system accounts in `/etc/shadow`.

Network controls

The network can be a source of many security concerns. You can do some simple checks to help ensure you're properly configured. First, display trusted hosts and users:

```
cat /etc/hosts.equiv
```

No trusted hosts should be allowed. You can delete this file. Next, display NFS files and parameters:

```
cat /etc/dfs/dfstab
```

Consult Practical UNIX and Internet Security for how to improve NFS security. Now, display the message of the day file:

```
cat /etc/motd
```

This file should contain a warning to unauthorized users stating that they aren't welcome.

Now, display unauthorized statement at login:

```
cat /etc/issue
```

Do the same to this file that you did for `/etc/motd`. Next, display the network services file:

```
cat /etc/inetd.conf
```

This file should contain services only used by your system. Comment out any unused services. Display the system accounts that aren't allowed to use FTP to transfer files:

```
cat /etc/ftpusers
```

This file should contain all system accounts including root. Display network services currently active:

```
rpcinfo -p
```

Make sure that it isn't running any processes that aren't needed (e.g., `rstatd`, `rusersd` and `rexed`). Now, display the version of `sendmail`:

```
#!/usr/ccs/bin/what /usr/lib/sendmail
```

It's always better to have the latest version of `sendmail` installed on your machine. For information about the most recent `sendmail` implementation, visit <http://www.sendmail.org/>.

Now, display the network rhost and netrc files with the following:

```
#find / -name .rhosts -ls
#find / -name .netrc -ls
```

These files don't have to exist on the system. To disable the user's ability to create `.rhosts` files, edit the `/etc/pam.conf` file (Solaris 2.6 and higher). If you can't get rid of these files, make sure that their permissions are 600 and a user in whose home directory they are located owns them. Now, display the user's profile file permissions for different shells:

```
#find / -name .profile -ls
#find / -name .login -ls
#find / -name .cshrc -ls
#find / -name .kshrc -ls
```

The user should be an owner of his profile. These files should only be readable by the owner.

Monitoring and logging

Logging can provide a wealth of useful security information if you set it up correctly. We can do this in a few steps. First, display the system events to log:

```
cat /etc/syslog.conf
```

Although this file is installed by default, its configuration should be adjusted to specify what messages are to be stored in what files or forwarded to another loghost on the local network.

By default, Solaris doesn't capture `syslog` events sent to `LOG_AUTH`. This information is very useful since it contains information on unsuccessful login attempts, successful and failed `su` attempts, reboots, and a wealth of other security-related information. Consult <http://www.cert.org/security-improvement/implementations/i041.08.html> for detailed syslogd configuration information. Now, display accounts that use the `su` command:

```
#cat /var/adm/sulog
```

Checking the `sulog` will tell you if your users are trying to become the root by searching for passwords. If you see dozens of `su` attempts from a particular user who isn't supposed to have access to the `root` account, you might want to ask him what he's trying to do.

Display audit events that have been defined:

```
cat /etc/security/audit_control
```

This file contains audit control information used by `auditd`. Note that the functionality of this file is available only if the Basic Security Module (BSM) has been enabled.

Display the following if the logging `cron` is enabled:

```
cat /etc/default/cron
```

The `CRONLOG` variable should be set to `YES`. Now, check for all failed login attempts with this:

```
cat /var/adm/loginlog
```

After five unsuccessful login attempts, all the attempts are logged in the `/var/adm/loginlog` file. By default this file doesn't exist, so no logging is done. To enable logging, create the `/var/adm/loginlog` file. Change permissions to 600. The owner of this file must be root. The group must be set to `sys`.

File and directory permissions

Files and directories can pose many security issues. Permissions need to be set to reflect your current permission policies to minimize the chances of system damage. First, display file permissions in the root directory:

```
ls -la /
```

Look for any unusual files in the root directory. Now, display file permissions in the `/etc` directory:

```
ls -la /etc
```

All the files in `/etc` should be kept unwritable by users other than root. Display file permissions in the `/etc/default` directory:

```
ls -la /etc/default
```

No files with write permissions are allowed in this directory. Now, display file permissions over system log files:

```
ls -la /var/adm
```

All files in this directory must be owned by system accounts (not actual human accounts) and not have world write permissions. Next, display file permissions for the scheduled files in the root crontab:

```
ls -l /var/spool/cron/crontabs/root
```

This file must have 400 permissions and be owned by root only. Display file permissions for the `cron` log file:

```
ls -l /var/cron/log
```

This file must have 600 permissions and must be owned by root. Display files owned by non-existent users or groups:

```
#find / \( -nouser -o -nogroup \) -ls
```

Delete these files or change the ownership for existing users and groups. Now, display `SUID` and `SGID` files owned by root:

```
#find / -user root \( -perm -4000 -o -perm  
=>-2000 \) -ls
```

It's a good idea to run this command soon after the system has been set up. Send the output to a file, and keep it for making comparisons later:

```
#find / -user root \( -perm -4000 -o -perm  
<SPAN&NBSP;CLASS='REF'>=>-2000 \) > files.check
```

Then once in a while run the following command:

```
#find / -user root \( -perm -4000 -o -perm  
<SPAN&NBSP;CLASS='REF'>=>-000 \) | diff - files.check
```

One of the ways to grant users the root privileges to do a specific job is by using the `sudo` application found at <http://smc.vnet.net/>. Now, display world-writeable files:

```
#find / -type f -perm -2 -ls
```

Most systems don't have any reason to have files that are writeable by anyone. Next, display world-writeable directories:

```
#find / -type d -perm -2 -ls
```

Review the output. You don't need it set for directories such as `/etc`, `/var`, `/dev`, `/devices` etc.

Other areas

You can find out the last time your system was rebooted with the following command:

```
last reboot /var/adm/wtmp
```

If you have accounts with a restricted shell, check to see if they're set up properly. The vanilla restricted shell `rksh` is breakable. To check whether restricted shell accounts are set up correctly, do the following:

- From the restricted shell account, start the `vi` editor.

- Inside of vi, set the following variable:
- `:set shell=/bin/sh`
- Type `:shell`. When you get a shell prompt, try `cd /`. If you succeed, your restricted shell account should be configured properly.

For how to configure a restricted shell account, refer to Practical UNIX and Internet Security.

It's always a good idea to check to see which kernel modules are loaded. Programs such as TTY Watcher, which can capture all users' keystrokes need to be loaded into the kernel. Make sure that no foreign modules are loaded. You can check which kernel modules are loaded by typing the following command:

```
modinfo
```

Conclusion

By using the operating system's commands, you can quickly check the security on your Solaris machine. You don't have to install and configure fancy security checking applications.

The security checklist provided in the article doesn't intend to be unique or complete. You can expand it with whatever your specific security needs may be.