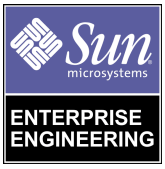
JumpStart™ Architecture and Security Scripts for the Solaris™ Operating Environment - Part 3

---

by Alex Noordergraaf - Enterprise Engineer

Sun BluePrints™ OnLine - September 2000



http://www.sun.com/blueprints

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303 USA
650 960-1300 fax 650 969-9131

Part No.: 806-6756-10
Revision 01, September 2000

JumpStart™ Architecture and Security Scripts for the Solaris™ Operating Environment - Part 3

Overview

This is the third and final article in a three part series discussing the JumpStart™ Architecture and Security Script (Toolkit) as a mechanism to secure Solaris™ Operating Environment (Solaris OE) systems.

The first article presented a detailed overview of the JumpStart product, and has step by step instructions for installing and configuring a JumpStart client and server, and the second article presented the configuration files and directories used by the Toolkit to harden Solaris OE systems.

The first article is available at:
`http://www.sun.com/blueprints/0700/jssec.pdf`

The second article is available at:
`http://www.sun.com/blueprints/0800/jssec2.pdf`

This article continues with an in-depth analysis of the configuration files, directories, and scripts used by the Toolkit to harden Solaris OE systems. It discusses all directories and their contents. A guide to adding new Toolkit functionality is also presented.

Supported Solaris Operating Environment Versions

The current release of the Toolkit works with the Solaris 2.5.1, 2.6, 7, and 8 OEs. Scripts which contain OS specific instructions will detect which version of the Solaris OE is being used, and will only run those tasks appropriate for the release

Installation

Along with the publication of this article the first public release of the Toolkit it being made available. This first public version of the Toolkit is version 0.1. Updates will be made to the Toolkit source separately from updates to this document. When downloading the Toolkit select the most recent copy. The instructions included below use filenames which are only correct for this release of the Toolkit:

Use the procedure below to download and install the Toolkit.

1. **Download the source file (`jass-0.1.tar.Z`).**

   The source file is located at

   `http://www.sun.com/blueprints/tools/jass`

2. **Extract the source file into the `/jumpstart` directory on the JumpStart server.**

   Use the `zcat` and `tar` commands as shown below:

```
# zcat jass-0.1.tar.Z | tar -xvf -
```

   Executing this command creates eight directories and their associated files. Based on your JumpStart server configuration, the Toolkit configuration information in the `/jumpstart/Drivers/user.init` will need modification.

Toolkit Architecture

There are eight directories in the Toolkit, as discussed in Part 2 of this series.

- `Drivers`
- `Files`
- `Finish`
- `OS`
- `Packages`
- `Patches`
- `Profiles`
- `Sysidcfg`

Each directory is discussed in more detail in the sections that follow. Where appropriate, each script, configuration file, or sub-directory is discussed individually. Suggestions are also made on how to modify and add additional scripts.

`Driver` Directory

The files in the `Driver` directory contains configuration information which specify which scripts will be run by the JumpStart server during client installation. The scripts called by the `/jumpstart/Driver` files are located in the `Finish` directory.

Driver Script Creation

All driver scripts have three parts:

The first part sets the directory path and calls the `driver.init` script. The `driver.init` script calls the `user.init`, which should contain all site-specific configuration information. Then, the driver.init will set those environment variables which are not site-specific and have not been defined by the `user.init`. All subsequent Toolkit scripts use the these environment variables.

The second part defines the FILES and SCRIPTS environment variables. Based on the definition of these variables, the `driver.run` script copies files to the JumpStart client and executes Finish scripts. The `FILES` variable defines those files which will be copied from the `Files` directory on the JumpStart server to the client. The

SCRIPTS variable defines what scripts will be executed during the installation of the client. Each of the Finish scripts available in the Toolkit will be discussed later in this article.

The final component to the architecture is the `driver.run` script. This script executes the contents of the FILES and SCRIPTS environment variables.

The following is an excerpt from a driver script showing the three parts.

```
DIR="‘/bin/dirname $0‘"

export DIR
. ${DIR}/driver.init

FILES="
               /etc/cron.d/cron.allow
               /etc/default/ftpd
               /etc/default/telnetd
"

SCRIPTS="
               account-removes.fin
               at-allow-create.fin
"
. ${DIR}/driver.run
```

Driver Script Listing

There are eight files in the `Drivers` directory. They are:

- `config.driver`
- `driver.init`
- `driver.run`
- `hardening.driver`
- `iplanet-enterprise-server.driver`
- `secure.driver`
- `user.init`
- `user.run`

The remainder of this section discusses these critical scripts in more detail.

config.driver

This driver script implements a mechanism to separate scripts which perform system configuration tasks from security specific scripts. Because of this separation mechanism, machines with different security requirements can still share the same base Solaris OE configuration driver.

Following is an excerpt from the `config.driver` script included with the Toolkit:

```
DIR="‘/bin/dirname $0‘"
export DIR

. ${DIR}/driver.init

FILES="
                /.cshrc
                /etc/inet/ntp.conf
                /etc/resolv.conf
"

SCRIPTS="
                set-root-password.fin
                set-term-type.fin
"

. ${DIR}/driver.run
```

This script performs several tasks. First, it calls `driver.init`. Then, it sets both the `FILES` and `SCRIPTS` environment variables. Once these environment variables are set the `driver.run` script is called. The `driver.run` script completes the execution of all configuration-specific scripts.

driver.init

The first script executed by any driver must be the `driver.init` script. The `driver.init` script sets the environment variables on which the Finish scripts depend. These variables are:

- `FILES_DIR`
- `FINISH_DIR`
- `JASS_SUFFIX`
- `PACKAGE_DIR`
- `PACKAGE_MOUNT`
- `PATCH_DIR`
- `PATCH_MOUNT`

- ROOT_DIR
- SI_CONFIG_DIR
- STANDALONE
- UNAME
- USER_DIR

Each of these variables were discussed in Part 2 of this series. For additional information on any of these variables, refer to the “Overview” section at the beginning of this article for a link to the earlier articles.

driver.run

This script is the core of the Toolkit. It takes all the information fed to it by earlier scripts and configuration files, then it:

- verifies the configuration;
- mounts the file systems to the JumpStart client;
- copies the files specified by the FILES environment variable;
- runs scripts specified by the SCRIPTS environment variable;
- unmounts the file systems from the JumpStart client.

Each of these functions are described in more detail below.

Verify Configuration

The first task of the driver.run script is verification of the Toolkit configuration by checking the following environment variables:

- FINISH_DIR
- PACKAGE_MOUNT
- PATCH_MOUNT
- UNAME

If these variables are not set, the verification process fails and the installation exits.

Mount Filesystems

Next the script calls an internal sub-routine called mount_filesystems. This routine mounts the following directories onto the JumpStart client:

- PACKAGE_MOUNT, which is mounted onto PACKAGE_DIR
- PATCH_MOUNT, which is mounted onto PATCH_DIR.

If other file system mounts points are required, user.run can be used to implement them.

Copy Files

After the mounts have completed successfully, the script copies over all files specified in the `FILES` environment variable (which can be set in any Finish script) to the JumpStart client. This copy mechanism is useful if many Solaris OE configuration files need to be replaced during a system installation.

Execute Scripts

After the previous scripts have been executed, the finish scripts listed in the `SCRIPTS` environment variable are executed in sequence. The output of these finish scripts are logged into the `/var/sadm/system/logs/finish.log` file on the JumpStart client. This is the standard log file used by any JumpStart command run on the client.

Unmount Filesystems

After all `Finish` scripts have been run, the `driver.run` script unmounts all filesystems mounted during “Mount Filesystems” process, then exits gracefully. At this point the JumpStart client reboots.

`hardening.driver`

All security specific scripts included in the Toolkit are listed in the `hardening.driver` script. This script, similar to the `config.driver` script, defines both files and scripts to be run by the `driver.run` script. Version 0.1 of the Toolkit implements all the recommendations made in the *Solaris Operating Environment Security BluePrint* referenced in the Bibliography, along with a few additional Solaris 8 OE specific scripts.

`iplanet-enterprise-server.driver`

This driver calls the `minimize-iplanet-enterprise-server.fin` script first presented in the onLine BluePrint article titled *Solaris Operating Environment Minimization for Security*. The script removes all Solaris packages not required to

successfully install and run the iPlanet™ Enterprise Server. The script has been updated to include support for the Solaris 8 OE in 32-bit mode. The following are the contents of the driver script:

```
DIR="‘/bin/dirname $0‘"
export DIR

. ${DIR}/driver.init

. ${DIR}/config.driver

SCRIPTS="
                minimize-iplanet-enterprise-server.fin
"
. ${DIR}/driver.run

. ${DIR}/hardening.driver
```

If a JumpStart client were to be built using this driver script it must be listed in the `rules` file for that JumpStart client. This script performs all the actions specified by the `config.driver` and `hardening.driver` scripts in addition to the minimization functionality in the `minimize-iplanet-enterprise-server.fin` script.

`secure.driver`

The following is the contents of the `secure.driver` script included with the Toolkit:

```
DIR="‘/bin/dirname $0‘"
export DIR

. ${DIR}/driver.init

. ${DIR}/config.driver

. ${DIR}/hardening.driver
```

This script is provided as a ready-to-use mechanism to implement all the hardening functionality in the Toolkit. This script performs the initialization tasks required, then calls the `config.driver` and `hardening.driver` scripts. This configures the system and performs all the hardening tasks specified in the `hardening.driver` script. The script should be the default script used in the `rules` file for the installation of clients.

user.init

This script provides a mechanism to specify user functions that will be used by the Toolkit. This script may override any of the default environment variables supplied by the Toolkit. This script should be used to add site-specific or organization-specific information to the Toolkit, minimizing future Toolkit migration issues.

This script provides default values for the `package_mount` and `patch_mount` environment variables. You must modify these variables to the specific JumpStart Server and directory path you will be using for each toolkit installation.

user.run

As with `user.init`, this script should be used to add any site-specific or organization-specific information into the Toolkit to avoid migration issues. The `user.run` script should contain all site-specific and organization-specific overrides for the `driver.run` script.

Files Directory

The `Files` directory is used in conjunction with the `FILES` environment variable and the `driver.run` script. This directory is used to store the files that will be copied to the JumpStart client.

The FILES Environment Variable and Files Directory Setup

The `FILES` environment variable is used to specify the complete Solaris OE path of files stored in the `/jumpstart/Files` directory. This environment variable can be used in the three following ways:

For example, the following is defined in the `hardening.driver` script:

```
FILES="
       /etc/ftpusers
"
```

By defining the FILES environment variable to include this file, the `/etc/ftpusers` file on the JumpStart client will be replaced by the `/jumpstart/Files/etc/ftpusers` file on the JumpStart server. Any file can be copied in this manner by simply including it in the `Files` directory and adding it to the `FILES` definition in the appropriate driver script.

The second option is to specify host-specific files by defining the `FILES` variable to contains something similar to following line:

```
/etc/syslog.conf.$HOSTNAME
```

In this scenario, the `/jumpstart/Files/etc/syslog.conf` files are only copied to a system with a hostname that matches *$HOSTNAME*. When there is both an `/etc/syslog.conf` and `/etc/syslog.conf.`*`$HOSTNAME,`* the host-specific file will have precedence.

The final option is to have the FILES variable specify a directory. When used, the entire directory contents is copied to the JumpStart client. If the FILES variable contains the following line:

```
/etc/rc2.d
```

then the entire contents of the `/jumpstart/Files/etc/rc2.d` directory on the JumpStart server will be copied to the JumpStart client.

Files Directory Listing

There are eleven files in the `Files` directory. They are:

- `/etc/ftpusers`
- `/etc/issue`
- `/etc/motd`
- `/etc/notrouter`
- `/etc/nsswitch.conf`
- `/etc/syslog.conf`
- `/etc/default/ftpd`
- `/etc/default/telnetd`
- `/etc/init.d/inetsvc`
- `/etc/init.d/nddconfig`
- `/etc/rc2.d/S70nddconfig`

The remainder of this section discusses these files in more detail.

/etc/ftpusers

The Solaris OE does not create an `ftpusers` file by default. The file included in the Toolkit contains entries for default system accounts including `root`, `daemon`, `sys`, `bin`, `adm`, `lp`, `smtp`, `uucp`, `nuucp`, `listen`, `nobody`, `noaccess`, and `nobody4`.

/etc/issue
/etc/motd

These files are based on US Government recommendations. They provide legal notice to users that their activities may be monitored.

/etc/notrouter

This file disables IP forwarding between interfaces on the system by creating an `/etc/notrouter` file. Once the JumpStart client is rebooted, the client will no longer function as a router, regardless of the number of network interfaces.

/etc/nsswitch.conf

This is an `nsswitch.conf` file configured so that a system will use DNS for name resolution. It is a copy of the `/etc/nsswitch.dns` shipped with Solaris 8 OE.

/etc/default/ftpd

This file enables the feature available in the Solaris 7 and 8 OEs to change the default FTP banner. The banner is changed by adding a `BANNER` entry to the `/etc/default/ftpd` file. The `/etc/default/ftpd` file included in the Toolkit creates a generic *Authorized Access Only* entry, which denies FTP version information to potential attackers.

/etc/default/telnetd

This file enables the feature available in Solaris 7 and 8 OEs to change the default TELNET banner. The banner is changed by adding the `BANNER` entry to the `/etc/default/telnetd` file. The `/etc/default/telnetd` file included in the Toolkit creates a generic *Authorized Access Only* entry, which denies TELNET version information to potential attackers.

/etc/init.d/nddconfig
/etc/rc2.d/S70nddconfig

These files copy over the `nddconfig` and `S70nddconfig` startup scripts required to implement the settings described in the *Solaris Operating Environment Network Settings for Security BluePrint*. See the Bibliography for the URL of this article.

/etc/init.d

This file replaces the default `/etc/init.d/inetsvc` with a minimized version containing only those commands required for the configuration of the network interfaces. The minimized script has only four lines as compared to the 256 lines of the Solaris 8 OE version. The minimized `inetsvc` script is as follows:

```
#!/bin/sh

/usr/sbin/ifconfig -au netmask + broadcast +
/usr/sbin/inetd -s -t &
```

Although this script has been used successfully by a variety of Sun customers, it has no support for DHCP. Therefore, this file should only be used in environments that use static IP addresses.

`Finish` Directory

The `Finish` directory contains the scripts which perform system modifications and updates during installation.

Finish Script Creation

Finish scripts run from a memory-resident mini-root running on the JumpStart client. The mini-root contains most of (but not all) the Solaris OE functions. When creating Finish scripts, it is sometimes necessary to execute commands using the `chroot` command.

To simplify portability and configuration issues, the environment variables defined in the `driver.init` script are used throughout the Toolkit. If additional variables are required they should be added as environment variables to the `user.init` and `user.run` scripts to avoid hard-coding specifics in the scripts.

Finish Script Listing

Each of the scripts in the Finish directory is briefly discussed in this section. The scripts fall into seven categories:

- Disable
- Enable
- Install
- Minimize
- Remove
- Set
- Update

Individual scripts in each category are discussed below. For additional background or justifications of the scripts see the previously published Sun BluePrints™ OnLine Security articles referenced in the Bibliography.

Note – You must view the actual Finish script to find out which modifications are being made.

Disable Finish Scripts

The following Disable finish scripts are discussed in this section:

- `disable-asppp.fin`
- `disable-autoinst.fin`
- `disable-automount.fin`
- `disable-core-generation.fin`
- `disable-dmi.fin`
- `disable-dtlogin.fin`
- `disable-keyserv-uid-nobody.fin`
- `disable-lp.fin`
- `disable-nfs-client.fin`
- `disable-nfs-server.fin`
- `disable-nscd-caching.fin`
- `disable-power-mgmt.fin`
- `disable-preserve.fin`
- `disable-remote-root-login.fin`
- `disable-rlogin-rhosts.fin`
- `disable-rpc.fin`
- `disable-sendmail.fin`
- `disable-slp.fin`
- `disable-snmp.fin`
- `disable-spc.fin`
- `disable-syslogd-listen.fin`

- disable-system-accounts.fin
- disable-uucp.fin

disable-asppp.fin

This script disables all the asppp startup and shutdown scripts (three kill scripts and one startup script) in the rc directories.

disable-autoinst.fin

This script disables the startup scripts used to re-initialize or re-install the system, including S30sysid.net, S71sysid.sys and S72autoinstall. These startup scripts will never be used in a JumpStart environment and should be disabled to prevent an intruder from reconfiguring the system.

disable-automount.fin

This script disables all the automounter startup and shutdown scripts. Five shutdown scripts and one startup script are disabled.

disable-core-generation.fin

This script disables the creation of core files by adding the appropriate command to the /etc/system file.

disable-dmi.fin

This script disables the DMI startup and shutdown scripts. Four shutdown scripts and one startup script are disabled.

disable-dtlogin.fin

This script disables all the CDE startup and shutdown scripts. One startup script and three shutdown scripts are disabled.

disable-keyserv-uid-nobody.fin

This script disables secure RPC access to user nobody by adding the -d option to the keyservd daemon startup command in the /etc/init.d/rpc file.

disable-lp.fin

This script disables all `lp` startup and shutdown scripts. There are a total of six scripts for the subsystems. Additionally, all `lp` access to the `cron` subsystem is removed by adding `lp` to the `/etc/cron.d/cron.deny` file and removing all `lp` commands in the `/var/spool/cron/crontabs` directory. This functionality is distinct from the `update-cron-deny.fin` script because the `lp` packages may or may not be installed on a system. In addition, the `lp` subsystem may be necessary while the functions removed by the `cron-deny-update.fin` script are not.

disable-nfs-client.fin

This script disables the NFS client startup scripts. Three kill scripts and one startup script are disabled.

disable-nfs-server.fin

This script disables the NFS server startup scripts. Seven kill scripts and one startup script are disabled.

disable-nscd-caching.fin

This script modifies the `nscd.conf` file to disable caching for `passwd`, `group`, and `hosts` by changing the value of the `enable_cache` option to `no` in the `/etc/nscd-caching.conf` file.

Note – Care should be taken when using the `disable-nscd-caching.fin` script in NIS and NIS+ environments, as `nscd` may be required.

---

disable-power-mgmt.fin

This script disables the auto power shutdown option on SPARC™ hardware platforms by creating a `/noautoshutdown` file. This script also disables the four scripts used for startup and shutdown of the `powerd` daemon.

disable-preserve.fin

This script disables the `/etc/init.d/PRESERVE` startup script.

disable-remote-root-login.fin

This script verifies the system is configured to disallow direct `root` logins. Even though this has been the default for the Solaris OE since the final update of 2.5.1, it should still be verified to ensure correct configuration.

disable-rlogin-rhosts.fin

This script disables `rhosts` authentication for `rlogin` by modifying the Pluggable Authentication Module (PAM) configuration in `/etc/pam.conf`.

disable-rpc.fin

This script disables the three kill and one startup scripts for Remote Procedure Calls (`RPC`).

disable-sendmail.fin

This script disables the `sendmail` daemon startup and shutdown script and adds an entry to the `cron` subsystem which executes `sendmail` once an hour. This method of purging outgoing mail is more secure than having the daemon running continually.

disable-slp.fin

This script disables all `Service Location Protocol (SLP)` startup and shutdown scripts. There are a total of four scripts for the subsystem.

disable-snmp.fin

This script disables the startup and shutdown scripts for the default Solaris OE SNMP daemons.

disable-spc.fin

This script disables all SunSoft™ Print Client (SPC) startup and shutdown scripts. There are a total of six scripts for the subsystem.

disable-syslogd-listen.fin

This script prevents the syslogd daemon from accepting SYSLOG messages from other systems on the network. This option has been added to version 8 of the Solaris OE, and is enabled by adding the -t option to the syslogd startup script. Even after using this option, processes on the system can still use syslogd.

disable-system-accounts.fin

This script disables system accounts and enables logging of access attempts. Disabled accounts are those with a UID of less then 100 or greater then 60,000 with the exception of root and sys. Access attempt logging is implemented by creating an /sbin/noshell script which denies access to the disabled account and logs the attempt (via SYSLOG) as an authentication error. Within the minimized Solaris OE, the logged accounts include daemon, bin, adm, lp, uucp, nobody, and noaccess.

disable-uucp.fin

This script disables the UUCP startup script.

Enable Finish Scripts

The following Enable finish scripts are discussed in this section:

- enable-32bit-kernel.fin
- enable-bsm.fin
- enable-ftp-syslog.fin
- enable-inetd-syslog.fin
- enable-priv-nfs-ports.fin
- enable-rfc1948.fin
- enable-stack-protection.fin

enable-32bit-kernel.fin

This script sets the boot-file variable in the EEPROM of Sun SPARC systems to the value of /kernel/unix. This forces the system to boot using a 32-bit kernel. It is useful for products that can run on the Solaris 7 OE or later, but must run in 32-bit only mode, such as Checkpoint's Firewall-1. This script is intended for sun4u systems.

enable-bsm.fin

This script performs all the necessary tasks involved in enabling the Basic Security Module (BSM) on a Solaris OE system in a lights-out data center environment. This includes:

- Running `bsmconv` script;
- removing the `L1A` (`STOP-A`) disable option which the `bsmconv` script added to `/etc/system;`
- editing the `/etc/security/audit_control` file created by `bsmconv;` and
- adding the `audit_warn` alias to the `sendmail` aliases file (if not there already).

After the system is rebooted, the BSM subsystem is enabled and logging is started.

enable-ftp-syslog.fin

This script forces the `in.ftpd` daemon to log all FTP access attempts through the `SYSLOG` subsystem. This option is enabled by adding the `-l` option to the `in.ftpd` command in the `/etc/inetd.conf` file.

enable-inetd-syslog.fin

This script enables logs of all incoming connection requests for service by the `inetd` daemon. When logging is enabled, `inetd` logs the source IP address, source TCP address, and service name through `SYSLOG`. Logging is enabled by adding the `-t` option to the `inetd` startup script in `/etc/init.d/inetsvc`.

enable-priv-nfs-ports.fin

This script sets the kernel variable `nfssrv:nfs_portmon` to 1, which restricts NFS requests to privileged ports only. After setting the variable in the `/etc/system` file, only NFS requests from ports less than 1024 are accepted.

enable-rfc1948.fin

This script enables RFC 1948 unique-per-connection ID sequence number generation by setting the `/etc/default/inetinit` `TCP_STRONG_ISS` value to 2.

enable-stack-protection.fin

This script the enables stack protection and logging included in all Solaris OE releases since version 2.6. These options are enabled by adding the following two commands to the /etc/system file:

- set noexec_user_stack = 1
- set noexec_user_stack_log = 1

After the two variables are set, the system denies attempts to execute the stack directly, and logs any stack execution attempt through SYSLOG. This facility is enabled to protect the system from common buffer overflow attacks.

Install Finish Scripts

The following Install finish scripts are discussed in this section:

- install-at-allow.fin
- install-cron-allow.fin
- install-fix-modes.fin
- install-loginlog.fin
- install-newaliases.fin
- install-openssh.fin
- install-recommended-patches.fin
- install-security-mode.fin
- install-strong-permissions.fin
- install-sulog.fin

install-at-allow.fin

This script restricts at command execution by creating an empty at.allow file in /etc/cron.d. An empty at.allow file forces the system to check the at.deny file for unauthorized at users. All users who require at access must now be added to the at.allow file. This script should be used in conjunction with the update-at-deny.fin script.

install-cron-allow.fin

This script creates a new /etc/cron.d/cron.allow file to restrict access to the cron subsystem. Only one account, root, is included in the new cron.allow file. No other system accounts are added. The root account will be the only account able to schedule tasks through the cron subsystem.

install-fix-modes.fin

This script both copies the `fix-modes` package (created by Casper Dik) from the JumpStart server to the client, and executes the script. You must first acquire the fix-modes package from:

```
ftp://ftp.wins.uva.nl/pub/solaris/fix-modes.tar.gz
```

compile it, and install it on the JumpStart server in `/jumpstart/Packages/FixModes.tar.Z`.

install-loginlog.fin

This script creates the `/var/adm/loginlog` file which is used by the system to log unsuccessful `login` attempts. The failed logins are logged after the number of failed logins has been exceeded. The number of failed logins permitted is specified in the `RETRIES` variable set in the `/etc/default/login` configuration file. See also `set-login-retries.fin`.

install-newaliases.fin

This script checks to see if the `/usr/bin/newaliases` file is present. If it isn't, and `/usr/lib/sendmail` is present, then it links `/usr/bin/newaliases` to `/usr/lib/sendmail`.

install-openssh.fin

This script is used to automate the installation of `OpenSSH` by installing the software and copying configuration information to the JumpStart client. Private and public key generation is performed during the installation. A compiled version of OpenSSH is required in the `/jumpstart/Packages` directory for this script to successfully complete.

OpenSSH is not included with the toolkit. It may be downloaded from:

```
http://www.openssh.com/
```

install-recommended-patches.fin

This script installs applicable patches from the /jumpstart/patches/*clustername* directory on the Jumpstart server. You must download and extract the *Recommended and Security Patch Clusters* to the /jumpstart/patches directory for the script to execute properly.

install-security-mode.fin

This script sets the Open Boot PROM security mode to command and sets the number of bad logins to zero. As it is not possible to script the setting of the EEPROM password during the JumpStart installation, it will have to been entered manually during the installation. Because this script requires human intervention it has been commented out of hardening.driver.

install-strong-permissions.fin

This script changes the permissions of the /etc/security directory to 0750 from the default value of 0755. By denying access to users not in the sys group, users have less access to information on the BSM subsystem. This script should be used in conjunction with the enable-bsm.fin script.

install-sulog.fin

This script creates the /var/adm/sulog file, which enables logging of all su attempts.

Minimize Finish Script

The following Minimize finish script is discussed in this section:

- minimize-iplanet-enterprise-server.fin

minimize-iplanet-enterprise-server.fin

This script implements the Solaris OE minimization procedure as described in the Sun BluePrints OnLine article *Solaris Minimization for Security: A Simple, Reproducible and Secure Application Installation Methodology.* The original script distributed with that article has been updated here for the 32-bit Solaris 8 OE and the Toolkit environment.

Remove Finish Script

The following Remove finish script is discussed in this section:

- `remove-unneeded-accounts.fin`

`remove-unneeded-accounts.fin`

This script removes unused Solaris OE accounts from the `/etc/passwd` and `/etc/shadow` files with the `passmgmt` command. This script removes the `smtp`, `nuucp`, `listen`, and `nobody4` accounts.

Set Finish Scripts

The following Set finish scripts are discussed in this section:

- `set-login-retries.fin`
- `set-rmmount-nosuid.fin`
- `set-root-password.fin`
- `set-system-umask.fin`
- `set-term-type.fin`
- `set-tmpfs-limit.fin`
- `set-user-password-reqs.fin`
- `set-user-umask.fin`

`set-login-retries.fin`

This script modifies the `RETRIES` variable in the `/etc/default/login` file to three from the default value of five. By reducing the logging threshold, additional information may be gained. The previously discussed `install-loginlog.fin` script enables the logging of failed login attempts.

`set-rmmount-nosuid.fin`

The default Solaris OE configuration allows `setuid` executable to work from removable media. After this script has modified the `/etc/rmmount.conf`, `setuid` executables on removable media will no longer execute with `setuid` privileges.

set-root-password.fin

This script automates setting the root password by setting the password to an initial value. The password used in this script should only be used during the installation and must be changed immediately after the JumpStart process has successfully completed. This script sets the root password to be ‘t00lk1t’.

Note – This script will only execute during a JumpStart software installation. It will not execute when the Toolkit is invoked from the command line.

---

set-system-umask.fin

This script creates startup scripts for each run level, which in turn, set the system `UMASK` properly to `022`. This script is not required for the Solaris 8 OE because the `CMASK` variable in `/etc/default/init` file performs this function.

set-term-type.fin

This script sets a default terminal type of `vt100` to avoid issues with systems not recognizing `dtterm`. This script is intended mainly for use on systems that do not have graphical consoles and are generally accessed over a terminal console or other serial link.

set-tmpfs-limit.fin

This script installs a limit on the disk space that can be used as part of a `tmpfs` filesystem. This limit can help prevent memory exhaustion. The usable space is limited by default in this script to 100 megabytes.

set-user-password-reqs.fin

This script enables more strict password requirements by enabling:

- Password aging
- Minimum intervals between password changes
- Increasing the password minimum length

This script is recommended for systems with non-privileged user access.

Note – Take care to ensure the `root` account is not inadvertently locked when running this script on restricted access servers.

`set-user-umask.fin`

This script adds an updated `UMASK` value of `077`, in the `/etc`, `/etc/skel`, and `/etc/default/login` files, and to the startup files for all default shells.

Note – A slightly less restrictive `UMASK` of 022 may be more appropriate for multi-user systems.

Update Finish Scripts

The following Update finish scripts are discussed in this section:

- `update-at-deny.fin`
- `update-cron-deny.fin`
- `update-inetd-conf.fin`

`update-at-deny.fin`

This script adds system accounts in `/etc/passwd` to the `/etc/cron.d/at.deny` file. Disabled accounts are those with a UID of less then 100 or greater then 60,000. When used in conjunction with the `install-at-allow.fin` file, no access will be permitted to the `at` subsystem.

`update-cron-deny.fin`

This script updates the `/etc/cron.d/cron.deny` file by adding the `sys`, `uucp`, `adm`, and `nobody4` system accounts to it. In addition, the `crontab` entries for `uucp` and `adm` are removed from the system `crontab`.

Depending on the packages installed, some modifications may be required to this Finish script because it has been written to run against minimized systems. This minimized system is described in the Sun BluePrints OnLine article, *Solaris Minimization for Security: A Simple, Reproducible and Secure Application Installation Methodology BluePrint*. In a minimized Solaris OE installation only the `uucp` and `admin crontab` entries need to be removed.

update-inetd-conf.fin

This script disables all standard entries in the `/etc/inetd.conf` file. The services are disabled after the script inserts a '#' at the start of each line. All services included in the base OS are disabled in Solaris OE versions 2.5.1 forward. Additional services installed by unbundled or third party software are not disabled.

OS Directory

This directory contains only Solaris OE images. These will be used by the JumpStart software installation process as the source of the client installation, and to provide the `add_install_client` and `rm_install_client` scripts which add new clients to the JumpStart environment. The installation naming convention recommended is `Solaris_`*`os version_2 digit month-2 digit year of CD release`*. The installation process documented in this article uses the Solaris 8 Operating Environment CD dated June 2000, so the directory name would be `Solaris_8.0_06-00`. By separating updates and releases of the Solaris OE, very fine control can be maintained for testing and deployment purposes.

Packages Directory

This directory contains software packages which can be installed with a Finish script. For example, the *iPlanet Enterprise Server* software package could be stored in the `Packages` directory so the appropriate Finish script can install the software as required.

Several Finish scripts are included in the Toolkit which perform software installation and basic configuration functions. Some of these functions were described in the preceding Finish Script section.

Patches Directory

This directory contains *Recommended and Security Solaris Patch Clusters*. Required clusters must be downloaded and extracted into this directory from `http://sunsolve.sun.com`. A directory should be created for each of the Solaris OE versions being used. There may be several directories including

2.5.1_Recommended and 2.6_Recommended within the Patches directory. These patch clusters are extracted in the Patches directory, which allows the patch installation script to run without having to extract the patch clusters for each system installation.

Profiles Directory

This directory contains all of the profiles. Profiles are files that contain configuration information used by the JumpStart software to determine what Solaris OE cluster to install (for example, Core, End User, Developer, or Entire Distribution), the disk layout to use, and the type of installation to perform (for example, standalone). These files are listed in the rules file to define how specific systems or groups of systems are built.

Profile Creation

The required and optional contents of profiles were discussed in Part 1 of this series. For additional information on profiles, refer to the *Profiles and Rules Creation* section of that article, which is listed in the Bibliography of this article.

Profile Configuration Files

A variety of profiles have been included with the Toolkit. These profiles are the standard JumpStart profiles. The profiles included in the Toolkit are:

- 32-bit-minimal.profile
- end-user.profile
- entire-distribution.profile

Most of the profiles supplied with the Toolkit have been customized for the lab environment in which the Toolkit was developed. Therefore, these profiles should be viewed as samples to be modified to suit the requirements of your site.

`Sysidcfg` Directory

This directory is used to store OS-specific versions of sysidcfg files. These files, as discussed in Part 1 of this series, are used to automate Solaris OE installations by providing the required information to the installation program. Because there is OE-specific information in these files in the Solaris 8 OE, a separate directory tree has been created to store that information.

Each Solaris OE has a separate directory and uses a naming scheme similar to that used by the `OS` directory. For each release there is a directory named: `Solaris_`*`OE Version`*. The Toolkit includes sample `sysidcfg` files for Solaris 2.5.1 through 8 which are in the following directories:

- `Solaris_2.5.1`
- `Solaris_2.6`
- `Solaris_7.0`
- `Solaris_8.0`

Conclusion

This article presented the first public release of the *JumpStart Architecture and Security Scripts Toolkit* which has been discussed in the two previous articles in this series. In addition to providing the download location for the Toolkit this article discussed the scripts in the Toolkit and their hardening, minimization, and configuration capabilities. Guidelines were also provided for adding new scripts. Recommendations on what Toolkit changes are required when moving to different JumpStart environments were also discussed. Additional information on JumpStart scripts are referenced in the Bibliography.

Bibliography

Solaris Advanced Installation Guide, Sun Microsystems,
`http://docs.sun.com`

Dik, Casper, *fix-modes tool*,
`ftp://ftp.wins.uva.nl/pub/solaris/fix-modes.tar.gz`

Noordergraaf, Alex , *JumpStart Architecture and Security Scripts for the Solaris Operating Environment - Part 1*, Sun BluePrints OnLine, July 2000,
http://www.sun.com/blueprints/0700/jssec.pdf

Noordergraaf, Alex, *JumpStart Architecture and Security Scripts for the Solaris Operating Environment - Part 2*, Sun BluePrints OnLine, August 2000,
http://www.sun.com/blueprints/0800/jssec2.pdf

Noordergraaf, Alex and Watson, Keith, *Solaris Operating Environment Minimization for Security*, Sun BluePrints OnLine, December 1999,
http://www.sun.com/blueprints/1299/minimization.pdf

Noordergraaf, Alex and Watson, Keith, *Solaris Operating Environment Network Settings for Security*, Sun BluePrints OnLine, December 1999,
http://www.sun.com/blueprints/1299/network.pdf

Noordergraaf, Alex and Watson, Keith, *Solaris Operating Environment Security*, Sun BluePrints OnLine, January 2000,
http://www.sun.com/blueprints/0100/security.pdf

Powell, Brad, et. al., *Titan Toolkit*,
http://www.fish.com/titan

Acknowledgements

I would like to thank: Glenn Brunette for his invaluable assistance in development and testing of the Toolkit; Keith Watson, Melodie Neal, Brad Powell, and Adrian Hiley for their review of the Toolkit documentation.

Glenn Brunette did a great deal of implementation on the Toolkit and was particularly helpful with the development of the configuration files, enhanced error checking, and command-line support in the Toolkit. Without his assistance the Toolkit would not be where it is today.

Keith Watson reviewed the end result of the Toolkit and made recommendations on needed changes, and also developed some elegant Finish scripts.

Melodie Neal, Brad Powell, and Adrian Hiley reviewed draft documentation on the Toolkit and provided many suggestions on how to improve it.

Author’s Bio: Alex Noordergraaf

Alex Noordergraaf has over nine years experience in the area of Computer and Network Security. As a Senior Security Architect in the Enterprise Engineering group of Sun Microsystems he is developing, documenting, and publishing security best practices through the Sun BluePrints OnLine program. Articles completed include recommendations on Solaris OE Security settings, Solaris OE Minimization, and Solaris OE Network settings.

Prior to his role in Enterprise Engineering he was a Senior Security Architect with Sun Professional Services where he worked with many Fortune 500 companies on projects that included Security Assessments, Architecture Development, Architectural Reviews, and Policy/Procedure review and development. In addition to providing billable services to customers he developed and delivered an Enterprise Security Assessment methodology and training curriculum to be used worldwide by SunPS. His customers have included major telecommunication firms, financial institutions, ISPs, and ASPs.