Deploying the Solaris Operating Environment Using a Solaris Security Toolkit CD

Steven Spadaccini, Enterprise Server Products

Sun BluePrints™ OnLine—September 2003

http://www.sun.com/blueprints

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95045 U.S.A.
650 960-1300

Part No. 817-3592-10
Revision 1.1, 10/8/03
Edition: September 2003

Please

Deploying the Solaris Operating Environment Using a Solaris Security Toolkit CD

This article describes how to construct a bootable Solaris™ Security Toolkit software CD, based on Sun's JumpStart framework, for building, configuring, and deploying new Solaris Operating Environment (Solaris OE) systems. This article is for intermediate and advanced users.

The Solaris Security Toolkit software is a collection of shell scripts combined to form a flexible and extensible framework for rapidly deploying hardened platforms running the Solaris OE. The Solaris Security Toolkit software is extremely versatile and can be used for much more than just hardening a system. By using the JumpStart framework and the Solaris Security Toolkit software, it is possible to rapidly deploy a hardened, secured system using a customized bootable CD with all the necessary installation, configuration, and patch cluster included.

This article contains the following topics:

Determining Business Case

The first task is to determine if building a customized, bootable Solaris Security Toolkit CD is right for your business case. We recommend that you consider the following:

- Does your business use JumpStart™ technology?
- Do you want to harden your systems in an automated way?
- Are there Service Level Agreements (SLAs) with your customers?

In some organizations, SLAs require the uptime of environments. In the case of a catastrophic failure, it might be impossible to fulfill a customer's SLA for a specified amount of time to bring the systems back online. Using a Solaris Security Toolkit CD, you can rebuild environments in a short amount of time and fulfill their SLAs to bring customer systems back online.

When building and deploying systems running the Solaris OE using JumpStart technology, it is generally assumed that the client system has the ability to remotely access the installation and configuration data using the networked file system (NFS) service. The JumpStart server might not be accessible for many reasons. Sun Professional Services (SunPS) works with customers in environments where a JumpStart server is not available, is not accessible via NFS, or is being upgraded. Also, some customers are not using JumpStart technology in any capacity and are installing or upgrading their systems manually. Although it is possible to resolve some of these issues and wait to install a system, there are cases when a system must be deployed in a relatively short amount of time using other methods.

If your organization and customers would benefit from using a customized bootable Solaris Security Toolkit CD, use the Solaris Security Toolkit framework to build a customized distribution of the Solaris OE. Administrators can customize or add additional files and configurations that are specific to an environment. During an installation from a Solaris 8 Installation CD, an administrator can install the distribution and perform minimal customizing. However, the customizing to the system might occur after the installation, such as package addition, package removal, and/or hardening. Also, an installation from the Sun Microsystems Solaris 8 OE CD does not take into account any system hardening. These issues can be automated by building a customized, bootable CD by utilizing the JumpStart and Solaris Security Toolkit framework. This method saves administrators time and provides them with the flexibility of deploying a complete, hardened system with your customizing.

Obtaining Support

The configurations for Sun systems implemented by the Solaris Security Toolkit software are Sun supported configurations. Support calls to Sun's support services are handled the same as other cases.

Note – The Solaris Security Toolkit software itself is not a supported Sun product. Sun's support services cannot accept calls about the Solaris Security Toolkit's scripts.

To obtain Solaris Security Toolkit software assistance and to submit bug reports, questions, suggestions, and feedback, use the Solaris Security Forum link at:

`http://www.sun.com/security/jass`

Feedback on how the Solaris Security Toolkit software works and words of encouragement to the developers are appreciated.

Creating a Build Specification

Before building a bootable CD, it is important to develop a build specification for the CD.

- A build specification contains all the packages, configurations, and files necessary to build an installation of the Solaris OE. Drivers and finish scripts used by the Solaris Security Toolkit software assist in deploying the build specification.

Note – This article deals with the Solaris 8 OE distribution only. You can use the technologies and processes described here for other distributions of Solaris OE. In this article, only the required packages (`SUNWCreq`) for a Solaris 8 OE distribution are used. It is possible that disk space for CD-R formatted disks could be an issue if you add optional packages or software to a CD. Therefore, we address only the required packages.

- Document the build specification.
- Establish policies within your organization. Make the policy available to all groups that require access and information to the build specification. Include as part of the policy a process for maintaining version control of the CD. Every modification to a distribution needs to be transferred onto the CD.

Building a Work Space

This section outlines the process for building a work space.

▼ To Build a Work Space

1. **Create a directory with an appropriate name.**

   This directory should be at least 1 gigabyte for your customized distribution. We recommend 2 gigabytes be available during development.

   For our example, we created a directory in `/export` called `core`.

2. **Transfer the data from the Sun Microsystems Solaris 8 media (Disk 1) onto the file system.**

   CODE EXAMPLE-1 illustrates the process. Alternatively, you can use the Volume Table of Contents (VTOC) from the Solaris media. For this alternate method, see CODE EXAMPLE-2.

   - To transfer data from media to the file system, perform the following:

CODE EXAMPLE-1 Transferring Data From Media to File System

```
# cd /cdrom/sol_8_1001_sparc/s0
# mkdir /export/core
# find . –print | cpio –pudm /export/core
# cd /export/core
# /etc/init.d/volmgt stop
# for i in 1 2 3 4 5
>do
>dd if=/dev/dsk/c0t6d0s${i} of=sol8.s${i} bs=512
>done
172800+0 records in
172800+0 records out
2560+0 records in
2560+0 records out
2560+0 records in
2560+0 records out
```

CODE EXAMPLE-1 Transferring Data From Media to File System *(Continued)*

```
# prtvtoc /dev/dsk/c0t6d0s0

/dev/dsk/c0t2d0s0 partition map

Dimensions:
   512 bytes/sector
   640 sectors/track
   1 tracks/cylinder
   640 sectors/cylinder
   2048 cylinders
   2048 accessible cylinders

Flags:
   1: unmountable
   10: read-only

Unallocated space:
   First     Sector    Last
   Sector     Count    Sector
   1224960      3200   1228159
   1237760     72960   1310719

                          First     Sector   Last
Partition  Tag  Flags     Sector     Count    Sector  Mount Directory
     0      4    10     01070720   1070719
     1      2    10     1070720    154240   1224959
     2      0    00     1224960    3200     1228159
     3      0    00     1228160    3200     1231359
     4      0    00     1231360    3200     1234559
     5      0    00     1234560    3200     1237759
# /etc/init.d/volmgt start
volume management starting
```

Note – Volume management must be stopped to execute the `prtvtoc` command on a CD. Volume management is then restarted after executing the `prtvtoc` command.

- To use the Volume Table of Contents (VTOC) from the Solaris media and the `dd` command, perform the following:

CODE EXAMPLE-2 Using VTOC to Transfer Media to File System

```
# dd if=/dev/dsk/c0t6d0s0 of=/export/core/sol8.cdrom.vtoc
>bs=512 count=1
>1
1+0 records in
1+0 records out
# /etc/init.d/volmgt start
```

Installing the Solaris Security Toolkit

This section outlines the steps for constructing the Solaris OE into a bootable CD-ROM for installation.

▼ To Install the Solaris Security Toolkit

1. **Inside the directory `s0`, remove all files and directories in the `.install_config` directory.**

   Later, after installing the Solaris Security Toolkit software, you will copy all contents into the `.install_config` directory.

2. **Download the Solaris Security Toolkit software from the following Web site:**

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

   We recommend that you use the tar format because extracting the file to the `.install_config` directory is more effective than using the Solaris package (pkg) format. For detailed information about downloading and installing the software, refer to the Sun BluePrint book *Securing Systems with the Solaris Security Toolkit*.

3. **Install the Solaris Security Toolkit software.**

CODE EXAMPLE-3 provides an example of the process for installing the Solaris Security Toolkit software.

CODE EXAMPLE-3 Installing Solaris Security Toolkit Software

```
# cp jass-4.0.tar.Z /export/core/s0/.install_config

# cd /export/core/s0/.install_config

# ls

jass-4.0.tar.Z

# uncompress jass-4.0.tar.Z

# tar -xvf jass-4.0.tar

x ./jass-4.0, 0 bytes, 0 tape blocks

x ./jass-4.0/CHANGES, 27397 bytes, 54 tape blocks

x ./jass-4.0/CREDITS, 3410 bytes, 7 tape blocks

x ./jass-4.0/Documentation, 0 bytes, 0 tape blocks

x ./jass-4.0/Documentation/BuildInf.pdf, 96402 bytes, 189 tape blocks

x ./jass-4.0/Documentation/audit_config.pdf, 128574 bytes, 252 tape blocks

x ./jass-4.0/Documentation/jass_config_install-v03.pdf, 144869 bytes, 283
tape blocks

x ./jass-4.0/Documentation/jass_internals-v03.pdf, 120116 bytes, 235 tape
blocks

x ./jass-4.0/Documentation/jass_quick_start-v03.pdf, 59085 bytes, 116 tape
blocks

x ./jass-4.0/Documentation/network-updt1.pdf, 97271 bytes, 190 tape blocks

# cd jass-4.0.0

# ls

add-client  Documentation   Finish         jass-execute   MANIFEST
Packages    QUESTIONAIRE    rules.SAMPLE   CHANGES        Drivers
INSTALL     LICENSE         nomatch.beg    Patches        README
Sysidcfg    CREDITS         Files

jass-check-sum  make-jass-pkg   OS           Profiles       rm-client
sysidcfg

# mv * /export/core/s0/.install_config

# cd /export/core/s0/.install_config

# rm –rf jass-4.0.0
```

4. **After you copy the Solaris Security Toolkit directory contents to the `.install_config` directory, copy any existing JumpStart installation profiles to the Profiles directory under `/export/core/s0/install_config`.**

   Be sure to place the files in the proper directories, as they were on the JumpStart server. For instance, any profiles should be contained in the Profiles directory.

Customizing the Solaris Security Toolkit

The Solaris Security Toolkit software runs a specified set of finish scripts during the build of the Solaris OE. This section describes how to customize the Solaris Security Toolkit software and a JumpStart profile based upon the build specification created earlier.

▼ To Customize the Solaris Security Toolkit

For detailed information about creating and customizing the Solaris Security Toolkit software and JumpStart profiles, refer to the following Sun BluePrint books:

- *Securing Systems with the Solaris™ Security Toolkit*
- *JumpStart™ Technology: Effective Use in the Solaris™ Operating Environment*

1. **Create your own profile, based upon your build specifications, requirements, and disk space.**

   The `toolkit.profile` gives the system information as to how the file system is to be built. Additionally, the `rules` file calls JumpStart finish scripts. The `rules` file provides instructions for which profile and/or driver should run during installation. CODE EXAMPLE-4 shows how to customize a sample `toolkit.profile`, created for this exercise.

CODE EXAMPLE-4 Customizing a JumpStart Profile

```
# cat toolkit.profile
install_type    initial_install
system_type     server
#
#       Assume 4Gb minimum rootdisk
#       1.5Gb root filesys, and 500Mb var
#       Swap based on physical memory
```

CODE EXAMPLE-4 Customizing a JumpStart Profile *(Continued)*

```
#
partitioning    explicit
filesys         rootdisk.s0     1500    /
filesys         rootdisk.s1     auto    swap
filesys         rootdisk.s3     500     /var
cluster         SUNWCreq
```

The following is an example of the `rules` file used for this exercise.

```
# more /export/core/s0/.install_config/rules
probe network
any     -       -       Profiles/toolkit.profile     Drivers/secure.driver
```

Additionally, when the Solaris OE is installed, the `sysidcfg` is read for any configuration information regarding time zone settings, language, and Internet protocol settings (IP). CODE EXAMPLE-5 is a sample `sysidcfg` file that is included with the Solaris Security Toolkit software distribution. During this exercise, we do not customize the `sysidcfg`.

CODE EXAMPLE-5 Sample `sysidcfg` File

```
# more sysidcfg
#
# Copyright (c) 2000-2002 by Sun Microsystems, Inc.
# All rights reserved.
#
#ident  "@(#)sysidcfg 2.5     02/02/21     SMI"
#


system_locale=en_US
timezone=US/Eastern
network_interface=primary {netmask=255.255.255.0
                           protocol_ipv6=no}
terminal=vt100
```

CODE EXAMPLE-5 Sample `sysidcfg` File *(Continued)*

```
security_policy=NONE

name_service=NONE

timeserver=localhost
```

Drivers are a collection of shell scripts that automate processes for installation options. Drivers call finish scripts, which perform tasks for installation and security hardening of a system. These custom-made drivers manipulate any files and/or security hardening requirements necessary for this build as designated by Solaris Security Toolkit software.

The first top-level driver that is called is the `secure.driver`. The `secure.driver` is a wrapper used to assist the automation of the finish scripts for system hardening. This driver calls additional drivers for system variable settings and specific, customized finish scripts.

The `driver.init` file sets variables necessary for the Solaris Security Toolkit software during the build of the Solaris OE and is not customized for this exercise. In case you want to review it, the `driver.init` file is located in the `.install_config` directory.

After the `secure.driver` sets all the necessary variables for it to perform its function, it calls the `config.driver` file. The `config.driver` captures configuration information about the system and is not intended to perform any security functions. The `config.driver` automates the installation of the patch cluster from the Patches directory located inside the `.install_config` directory. It is necessary to apply the security patches before any hardening takes place on a system. Patches have a history of enabling services that might have been previously disabled. Therefore, the `config.driver` patches the system before any hardening is applied to a system.

2. **Create and modify a `user.run` file.**

   A sample `user.run.SAMPLE` file is included in the `.install_config` directory and is used for this exercise. The `user.run` file tries to mount an NFS file system to install the patch cluster. Because we are building a Solaris OE from CD, we need to tell the Solaris Security Toolkit software to find the patch cluster locally. Therefore, we replace NFS with LOFS (shown in bold) in the `user.run` file. LOFS tells the software that the patch cluster is on the loopback file system, as opposed to being remotely accessed using NFS. The change is shown in CODE EXAMPLE-6.

CODE EXAMPLE-6 Modifying the `user.run` File

```
#
# cat user.run
# !/bin/sh
#
```

CODE EXAMPLE-6 Modifying the `user.run` File *(Continued)*

```
#
# Copyright (c) 2002 by Sun Microsystems, Inc.
# All rights reserved.
#
#ident  "@(#)user.run.SAMPLE 3.2     02/08/30     SMI"
#
# This file is to be used to override or specify user functions that
# will be used by this programs using this toolkit.  Note that this
# script is called within the "driver.run" and so will override any
# of the default functions supplied by this toolkit.  In general,
# this script should be modified as it will not be replaced during
# upgrades.

mount_fs()
{
   # This is a "helper" function for mount filesystems.  This generally
   # will not be called directly.

   # Parameters to this function are:
   # $1     = Server and fully qualified path to mount.
   # $2     = Name of the mount point.

   cd /

   if [ "${1}" = "" ]; then
      logError "The server and path to be mounted were not specified."
      return
   fi

   if [ "${2}" = "" ]; then
      logError "The mount point was not specified."
      return
   fi

   if [ ! -d ${2} ]; then
      mkdir -p ${2}
   fi
#
   echo "NOTICE: Mounting ${1} on ${2}."
   mount -F lofs ${1} ${2}
}
                    .
                    .
                    .
```

Now when the `config.driver` is run, the `hardening.driver` is used to harden the Solaris OE. The `hardening.driver` is built to the specifications necessary to function in the environment and is not modified for this exercise.

3. **Modify the Solaris Security Toolkit `user.init` file as shown in** CODE EXAMPLE-7**.**

CODE EXAMPLE-7 Modifying the user.init File

```
#
# !/bin/sh
# Copyright (c) 2002 by Sun Microsystems, Inc.
# All rights reserved.
#
#ident  "@(#)user.init.SAMPLE 3.2     02/08/30     SMI"
#
# This file is to be used to override or specify user variables that will
# be used by the scripts in this toolkit.  Note that this script is
# called before both "driver.init" and "finish.init" so any changes made
# in this file will not be overwritten.
#
# In general, this script should be modified as it should not be
# replaced during upgrades.
#
# For more information on how to extend or enhance variables used by the
# Finish scripts, refer to the "finish.init" script.
#
#
**********************************************************************
*****

JASS_SERVER="`df -k /cdrom | tail -1 | awk -F: '{ print $1 }'`"
export JASS_SERVER
JASS_PACKAGE_MOUNT=/tmp/install_config/Packages
JASS_PATCH_MOUNT=/tmp/install_config/Patches
#
if [ "${JASS_PACKAGE_MOUNT}" = "" ]; then
   JASS_PACKAGE_MOUNT="${JASS_SERVER}:/jumpstart/Packages"
   # echo "JASS_PACKAGE_MOUNT is set to ${JASS_PACKAGE_MOUNT}."
fi

export JASS_PACKAGE_MOUNT
                        .
                        .
                        .
```

During the build of the system, the installation expects to be explicitly told where the JASS_PACKAGE_MOUNT and JASS_PATCH_MOUNT are exported. This modification is necessary for this exercise because the Solaris Security Toolkit software mounts the Packages and Patches from /jumpstart/Packages or /jumpstart/Patches, respectively. The file should be in the Drivers directory within the .install_config directory. A sample user.init file is included with the Solaris Security Toolkit software distribution.

Building a Customized, Bootable Installation

Now we are ready to create the image of the customized Solaris OE. Two packages, SUNWcdrw and SUNWmkcd, are required to assist in this process.

▼ To Build a Customized, Bootable Installation

1. **Use the** pkginfo **command to check your system for the required packages, as shown in the following example**:

```
# pkginfo SUNWcdrw SUNWmkcd
system      SUNWcdrw       utility for writing to CD-R/RW disks
system      SUNWmkcd       CD creation utilities
```

2. **Use the** mkisofs **command to create the image by combining the VTOC and slices, as shown in** CODE EXAMPLE-8**.**

   By using the mkisofs command, which is part of the SUNWcdrw package, you can create your CD image.

CODE EXAMPLE-8 Creating the CD Image

```
 # mkisofs -R -d -L -l -o ${BASEDIR}/sol8.s0 ${BASEDIR/s0
                   .
                   .
                   .
 Total extents actually written = 233153
 Total translation table size: 0
 Total rockridge attributes bytes: 3987542
 Total directory bytes: 22243328
```

CODE EXAMPLE-8 Creating the CD Image *(Continued)*

```
# mkisofs –R –d –L –l –o ${BASEDIR}/sol8.s0 ${BASEDIR/s0
Path table size(bytes): 158286
Max brk space used 159a000
233153 extents written (455 Mb)

# dd if=sol8.s0 of=/export/core/new.sol8.s0 bs=512 skip=1

1065599+0 records in
1065599+0 records out
```

3. **Add `1` to the sectors for slice 0 to pad the VTOC, so that it functions correctly with the CD.**

   The VTOC needs to be padded correctly to function with the CD. Therefore, the pad is created by adding 1 to the sectors for slice 0 (s0) and having the sum subtracted by the number of sectors in the original (unmodified) slice 0 of the CD, as shown with the `prtvtoc` command in CODE EXAMPLE 9.

CODE EXAMPLE 9 Padding the VTOC

```
# bc
 1070720 - (1065599+1))
5120
# dd if=/dev/zero of=pad.s0 bs=512 count=3840
  5120+0 records in
  5120+0 records out
# cd /export/core
# cat sol8.cdrom.vtoc new.sol8.s0 pad.s0 sol8.s1 sol8.s2 sol8.s3 sol8.s4
sol8.s5 >
   /export/core/cd.image
# cdrw –d cdrom0 –I cd.image
  Initializing the device……done.
  Done.
  Finalizing (can take up to 4 minutes)..done.
```

We created a script to help automate the creation of the bootable CD image. The script in CODE EXAMPLE-10 includes the commands that were listed in the previous code examples in this section. The script provides error messages based upon incorrect sizing of the slices for the CD.

CODE EXAMPLE-10 Script Sample for Building a Bootable Installation CD

```
#
#>.. !/bin/ksh -x
BASEDIR=`pwd`
SLICE_0_SIZE=1070720
IMAGENAME=sol8cd.image
```

CODE EXAMPLE-10 Script Sample for Building a Bootable Installation CD *(Continued)*

```
#



if [ ! -d s0 ]; then
        echo "Cannot continue: s0 directory does not exist."
        exit 1
fi

rm ${BASEDIR}/${IMAGENAME} ${BASEDIR}/pad.s0 ${BASEDIR}/new.sol8.s0
${BASEDIR}/sol8.s0

mkisofs -R -d -L -l -o ${BASEDIR}/sol8.s0 ${BASEDIR}/s0

FSSIZE=$(dd if=${BASEDIR}/sol8.s0 of=${BASEDIR}/new.sol8.s0 bs=512 skip=1
2>&1 | \
        head -1 | cut -d+ -f1 )

echo $FSSIZE

PADBLOCKS=$(( $SLICE_0_SIZE - ($FSSIZE + 1) ))

if [ $PADBLOCKS -lt 1 ]; then
        echo "Cannot continue: Slice 0 is too large."
        echo "Need to remove an estimated $(( $PADBLOCKS / 2 ))k of files"
        exit 1
fi
dd if=/dev/zero of=${BASEDIR}/pad.s0 bs=512 count=$PADBLOCKS

cd ${BASEDIR}
cat sol8.cdrom.vtoc new.sol8.s0 pad.s0 sol8.s1 sol8.s2 sol8.s3 sol8.s4
sol8.s5 > ${IMAGENAME}
```

If you run this script, it creates a `cd.image` file. This file contains all the necessary information needed to build a core Solaris OE. Therefore, if no errors are reported during the creation of the CD, your image is ready for installation. Typically, errors that occur during the creation of an image are caused by an incorrect pad size and / or a slice being too large.

4. **Write the CD image onto disk using the following command:**

```
# cdrw -d cdrom0 -i /export/core/cd.image
```

5. **After the image is written to CD, test the CD image by using the following command to boot the CD from the OK prompt on your test client Solaris OE system:**

CODE EXAMPLE-11 Testing the CD

```
ok boot cdrom - install
Resetting ...

Sun Ultra 5/10 UPA/PCI (UltraSPARC-IIi 270MHz), No Keyboard
OpenBoot 3.31, 128 MB (50 ns) memory installed, Serial #11024924.
Ethernet address 8:0:20:a8:3a:1c, Host ID: 80a83a1c.

Rebooting with command: boot cdrom - install
Boot device: /pci@1f,0/pci@1,1/ide@3/cdrom@2,0:f  File and args: - install
SunOS Release 5.8 Version Generic_108528-11 64-bit
Copyright 1983-2001 Sun Microsystems, Inc.  All rights reserved.
Configuring /dev and /devices
Type  'go' to resume
ok boot cdrom - install
Resetting ...

Sun Ultra 5/10 UPA/PCI (UltraSPARC-IIi 270MHz), No Keyboard
OpenBoot 3.31, 128 MB (50 ns) memory installed, Serial #11024924.
Ethernet address 8:0:20:a8:3a:1c, Host ID: 80a83a1c.

Rebooting with command: boot cdrom - install
Boot device: /pci@1f,0/pci@1,1/ide@3/cdrom@2,0:f  File and args: - install
SunOS Release 5.8 Version Generic_108528-11 64-bit
Copyright 1983-2001 Sun Microsystems, Inc.  All rights reserved.
Configuring /dev and /devices
Using RPC Bootparams for network configuration information.
Skipping interface hme0
Searching for configuration file(s)...
The system is coming up.  Please wait.
in.rdisc: No interfaces up

The Solaris Installation Program
                      .
                      .
                      .

Starting Solaris installation program...
Searching for JumpStart directory...
 <<< Using CD Default >>>
Checking rules.ok file...
Using profile: Profiles/toolkit.profile
```

CODE EXAMPLE-11 Testing the CD *(Continued)*

```
Using finish script: Drivers/secure.driver
Executing JumpStart preinstall phase...
Searching for SolStart directory...
Checking rules.ok file...
Using begin script: install_begin
Using finish script: patch_finish
Executing SolStart preinstall phase...
Executing begin script "install_begin"...
Begin script install_begin execution completed.

Processing default locales
- Specifying default locale (en_US)

Processing profile
- Selecting cluster (SUNWCreq)

Installing 64 bit Solaris packages
-  Selecting cluster (SUNWClux)
                        .
                        .
                        .
Verifying disk configuration
- WARNING: Changing the system's default boot device in the EEPROM

Verifying space allocation
- Total software size:  198.24 Mbytes

Preparing system for Solaris install
Configuring disk (c0t0d0)
- Creating Solaris disk label (VTOC)

Creating and checking UFS file systems
         - Creating / (c0t0d0s0)
         - Creating /var (c0t0d0s3)
         - Creating /opt (c0t0d0s6)

Beginning Solaris software installation

Starting software installation
        SUNWkvmx.u..done.  197.91 Mbytes remaining.
                           .
                           .
                           .

Completed software installation
```

CODE EXAMPLE-11 Testing the CD *(Continued)*

```
Solaris 8 software installation succeeded

Customizing system files
        - Mount points table (/etc/vfstab)
        - Network host addresses (/etc/hosts)

Cleaning devices

Customizing system devices
        - Physical devices (/devices)
        - Logical devices (/dev)

Installing boot information
        - Installing boot blocks (c0t0d0s0)
        - Updating system firmware for automatic rebooting

Installation log location
        - /a/var/sadm/system/logs/install_log (before reboot)
        - /var/sadm/system/logs/install_log (after reboot)

Installation complete
Executing SolStart postinstall phase...
Executing finish script "patch_finish"...

Finish script patch_finish execution completed.
Executing JumpStart postinstall phase...
Executing finish script "Drivers/secure.driver"...

================================================================
secure.driver: Driver started.
================================================================
                      .
                      .
                      .
================================================================
secure.driver: Starting finish script: install-recommended-
patches.fin
================================================================

Checking installed patches...
Verifying sufficient filesystem capacity (dry run method)...
Installing patch packages...

                     .
                     .
                     .
```

CODE EXAMPLE-11 Testing the CD *(Continued)*

```
Resetting ...

Sun Ultra 5/10 UPA/PCI (UltraSPARC-IIi 270MHz), No Keyboard
OpenBoot 3.31, 128 MB (50 ns) memory installed, Serial #11024924.
Ethernet address 8:0:20:a8:3a:1c, Host ID: 80a83a1c.

Re-enter your root password.

Press Return to continue.

System identification is completed.
syslog service starting.
The system is ready.

|-------------------------------------------------------------|
| 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.       |
|-------------------------------------------------------------|

unknown console login:
```

About the Author

Steven Spadaccini has over eight years of experience in network security and system administration. Currently, he is a member of Enterprise Server Products and High-end Software as a security engineer on next-generation enterprise systems. As a former member of Sun Professional Services, Steven served as a security engineer for financial and non-financial areas conducting security assessments, implementations, and architecture design and review.

Acknowledgements

Special thanks to Stephen Putre, Glenn Brunette, and Alex Noordergraaf for providing technical content and guidance throughout the creation of this article.

Related Resources

Publications

- Howard, John S. “Building a Bootable JumpStart Installation CD-ROM,” Sun BluePrints OnLine, March 2001, `http://www.sun.com/solutions/blueprints/0301/BuildBoot.pdf`.
- Howard, John S. and Noodergraaf, Alex. *JumpStart™ Technology: Effective Use in the Solaris™ Operating Environment*, The Official Sun Microsystems Resource Series, Prentice Hall, October 2001.
- Howard, John S. and Noordergraaf, Alex. “Customizing JumpStart Framework for Installation and Recovery,” Sun BluePrints Online, August 2002, `http://www.sun.com/solutions/blueprints/0802/816-7587-10.pdf`.
- Noordergraaf, Alex. “Solaris Operating Environment Security: Updated for the Solaris 9 Operating Environment,” Sun BluePrints OnLine, December 2002, `http://www.sun.com/solutions/blueprints/1202/816-5242.pdf`.
- Noordergraaf, Alex and Brunette, Glenn. *Securing Systems with the Solaris Security Toolkit*, Sun Microsystems, Prentice Hall Press, ISBN 0-13-141071-7, June 2003.

- Noordergraaf, Alex, et. al. *Enterprise Security: Solaris Operating Environment Security Journal, Solaris Operating Environment Versions 2.5.1, 2.6, 7, and 8*, Sun Microsystems, Prentice Hall Press, ISBN 0-13-100092-6, June 2002.
- *Solaris 8 Advanced Installation Guide*, February 2002, `http://docs-pdf.sun.com/816-2411/816-2411.pdf`.

Web Sites

Note – Sun is not responsible for the availability of third-party Web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused by or in connection with the use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

- List open files software, `ftp://vic.cc.purdue.edu/pub/tools/unix/lsof/`
- Sun BluePrints OnLine, `http://sun.com/blueprints`
- Solaris Security Toolkit software, `http://sun.com/blueprints/tools/`
- SunSolve, `http://sunsolve.sun.com`

Ordering Sun Documents

The SunDocs[SM] program provides more than 250 manuals from Sun Microsystems, Inc. If you live in the United States, Canada, Europe, or Japan, you can purchase documentation sets or individual manuals through this program.

Accessing Sun Documentation Online

The `docs.sun.com` web site enables you to access Sun technical documentation online. You can browse the `docs.sun.com` archive or search for a specific book title or subject. The URL is `http://docs.sun.com/`

To reference Sun BluePrints OnLine articles, visit the Sun BluePrints OnLine Web site at: `http://www.sun.com/blueprints/online.html`.