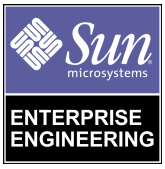
Solaris™ Operating Environment Minimization for Security: A Simple, Reproducible and Secure Application Installation Methodology

---

By Alex Noordergraaf and Keith Watson - Global Enterprise Seurity Service

Sun BluePrints™ OnLine - December 1999



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.: 806-4050-10
Revision 01, December 1999

Please
Recycle

Solaris™ Operating Environment Minimization for Security: A Simple, Reproducible and Secure Application Installation Methodology

> System penetration by outsiders [has] increased for the third year in a row; 30% of respondents report intrusions. Those reporting their Internet connection as a frequent point of attack rose for the third straight year: from 37% of respondents in 1996 to 57% in 1999. Meanwhile, unauthorized access by insiders also rose for the third straight year; 55% of respondents reported incidents. …financial loses … mounted to over $100,000,000.
>
> - Computer Crime and Security Survey, 1999

How to secure computer systems against unauthorized access is one of the most pressing issues facing today's data center administrators. Recent data suggests that the number of unauthorized access continues to rise, as do the monetary losses associated with these security breaches.

One way to reduce system vulnerabilities is to minimize the amount of software on a server. Fewer software components on a server means fewer security holes to detect and fill. The majority of system penetrations are accomplished through the exploitation of security holes in the operating system itself. Thus, minimizing the number of operating system (OS) modules installed on a server can greatly improve overall system security by reducing the sheer number of vulnerabilities.

This article, the first in a series of Sun BluePrints™ OnLine discussing securing data centers, focuses on operating system installation practices designed to improve overall system security by proposing a minimized and automated Solaris Operating Environment installation methodology. Unfortunately, the minimal OS requirements of a server vary depending on the applications, operating system release being

utilized, and the hardware itself. The process presented in this article can be used to assist the reader in determining the minimum OS modules which must be installed on a particular server. This is done within the framework of a Solaris Jumpstart™ installation, which makes it possible to completely automate the installation process. This automation is particularly important in a data center environment, where machines typically number in the hundreds.

Background

The Solaris Operating Environment installation process requires the selection of one of four *installation clusters*:

- *Core*
- *End User*
- *Developer*
- *Entire Distribution*

Each installation cluster represents a specific group of *packages* (operating system modules) to be installed. This grouping together of packages into large clusters is done to simplify the installation of the OS for the mass market. Because each of these installation clusters contains support for a variety of hardware platforms (Solaris™ Operating Environment (Intel Platform Edition), microSPARC™, UltraSPARC™, UltraSPARC II, and so on) and software requirements (NIS, NIS+, DNS, OpenWindows™, Common Desktop Environment (CDE), Development, CAD, and more), far more packages are installed than will actually ever be used on a single Solaris Operating Enironment.

The *Core* cluster installs the smallest Solaris Operating Environment image. Only packages that may be required for any SPARC™ or Solaris Operating Environment (Intel Platform Edition) system are installed. The *End User* cluster builds on the *Core* cluster by also installing the window managers included with the Solaris Operating Environment (OpenWindows and CDE). The *Developer* and *Entire Distribution* clusters include additional libraries, header files, and software packages that may be needed on systems used as compile and development servers.

The size of the clusters varies significantly: the *Core* cluster contains only 39 packages and uses 52MBytes; the *End User* cluster has 142 packages and uses 242 MBytes; the *Developer* cluster has 235 packages and consumes 493 MBytes of disk space. Experience to date has shown that in many cases, a secure server may require only 10 Solaris Operating Environment packages and use as few as 36MBytes of disk space.

Installing unnecessary services, packages, and applications can severely compromise system security. One well known example of this is the `rpc.cmsd` daemon, which is unnecessary on many data center systems. This daemon is installed and started by default when the *End User*, *Developer*, or *Entire Distribution* cluster is chosen during the installation process.

There have been many bugs filed against the rpc.cmsd subsystem of OpenWindows/CDE in the last few years, and at least two CERT advisories (CA-99-08, CA-96.09). To make matters even worse, scanners for `rpc.cmsd` are included in the most common Internet scanning tools available on the Internet. The best protection against `rpc.cmsd` vulnerabilities is to not install the daemon at all, and avoid having to insure it is not accidentally enabled.

The problem described above is well known in the computer industry, and there are hundreds of similar examples. Not surprisingly, almost every security reference book ever written discusses the need to perform “minimal OS installations” [Garfinkel]. Unfortunately, this is easier said than done. Other than the occasional firewall, no software applications are shipped with lists of their package requirements, and there’s no easy way of determining this information other then through trial and error.

Because it is so difficult to determine the minimal set of necessary packages, system administrators commonly just install the *Entire Distribution* cluster. While this may be the easiest to do from the short-term perspective of getting a system up and running, it makes it nearly impossible to secure the system. Unfortunately, this practice is all too common, and is even done by so-called experts brought in to provide infrastructure support, web services, or application support. (If your organization is outsourcing such activities, be sure to require the supplier to provide information on what their OS installation policies and procedures are, or you may be in for some unpleasant surprises.)

The rest of this article presents one method for determining the minimal set of packages required by a particular application—the iPlanet™ Enterprise Server. Future articles will discuss other applications. The tentative list includes NFS™ Servers (with SecureRPC and Solstice DiskSuite™), iPlanet™ WebTop, and Sun™ Cluster. If you have followed this procedure and developed the scripts for a particular application, please forward them to the authors for inclusion in future articles.

Internal Security Procedures

When presenting these recommendations as a framework for use on both internal and external servers there is, on occasion, some resistance. Arguments frequently include such myths as “hackers only come from the Internet” or “our employees are trustworthy” and that precautions need only be taken on external systems.

Unfortunately these beliefs are contradicted by reports generated by the FBI. The following is an excerpt from a statement by Michael A. Vatis Director, NIPC of the FBI:

> The disgruntled insider is a principal source of computer crimes … 55% of respondents reported malicious activity by insiders.
>
> There are many cases in the public domain involving disgruntled insiders. For example, Shakuntla Devi Singla used her insider knowledge and another employee's password and logon identification to delete data from a U.S. Coast Guard personnel database system. It took 115 agency employees over 1800 hours to recover and reenter the lost data. Ms. Singla was convicted and sentenced to five months in prison, five months home detention, and ordered to pay $35,000 in restitution.
>
> In another case, a former Forbes employee named George Parente hacked into Forbes systems using another employee's password and login identification and crashed over half of Forbes' computer network servers and erased all of the data on each of the crashed services. The data could not be restored. The losses to Forbes were reportedly over $100,000.
>
> -NIPC Cyber Threat Assessment

Based on these statistics most attacks actually originate from within an organization. Hence, just as many precautions must be taken when building internal systems as external systems. In light of the sensitive information internal systems normally contain, they should be protected even more carefully then external systems.

Test Environment

The processes presented in this methodology are oriented toward the classic lights-out data center environment. The following assumptions were made about server configurations:

- JumpStart™ software is available for system installations.
- JumpStart software has been configured properly for hands-off system installation and configuration (see References for additional information).
- Terminal consoles (character based) are used for console access.
- No video cards will be installed on any of the systems.
- No X Window server software will be required on the server.

The software builds were performed on sun4m-based systems which are microSPARC, SuperSPARC™, and hyperSPARC™ based (SPARC 5, 20, 10, Classic, etc.) and use SBUS interface cards. Additional packages will be required to support other hardware platforms.

Methodology Overview

The goal of this effort is to create a simple, reproducible, and secured application installation methodology. A secondary benefit is the automation of the entire operating system and software installation process.

The basic steps are outlined below. Each step is then further explained in the sections that follow.

1. Verify that JumpStart software is using the latest Solaris Operating Environment release.
2. Install the *Core Solaris* Operating Environment cluster plus any additional required packages.
3. Install all patches, including:
   a. recommended Solaris Operating Environment and Security patches.
   b. patches recommended by the product.
4. Remove all unnecessary packages.
5. Use JumpStart software to configure the OS for the datacenter environment.This includes specifying:
   a. network specifics such as default router, IP address, etc.
   b. OS configurations including name resolution, time server, etc.
   c. appropriate logging and auditing levels.
6. Install and configure the software package.
7. Check the logs for errors; if necessary, fix the errors and repeat the installation process.
8. Test the software installation.

1. Verify that JumpStart Software is Using the Latest Solaris Operating Environment Release

Verify Solaris Operating Environment release installed on the JumpStart boot server. For the purposes of the testing performed for this article the following Solaris Operating Environment revisions were used:

- Solaris 2.5.1 (11/97)

- Solaris 2.6 (5/98)
- Solaris 7 (8/99)

The installation and configuration of a Jumpstart server is beyond the scope of this article. Please see the "References" for pointers to both an excellent FAQ available from SunSolve™ and the Solaris Jumpstart manual available at http://docs.sun.com.

The scripts used in the validation and testing of this methodology are linked to "Appendix A: Scripts". Only those scripts specific to the iPlanet Enterprise Server installation are included, but all are based on examples found in the Solaris Operating Environment Advanced System Installation manual available from http://docs.sun.com. The iPlanet scripts included are:

- `bp-iplanetes.profile`: defines the cluster and packages to be installed by Jumpstart server.
- `bp-iplanetes.driver`: provides a framework in which a number of other scripts can be run.
- `bp-iplanetes-pkg-rm.fin`: removes unnecessary Solaris Operating Environment packages.
- `bp-iplanetes.fin`: extracts and installs the iPlanet software onto the server.

2. Install the *Core Solaris* Operating Environment Cluster Plus any Required Packages

The initial installation should only include the *Core Solaris* Operating Environment cluster and a few other packages which contain critical functionality. In Jumpstart server terminology, this is referred to as the SUNWCreq cluster. The following packages were required for the iPlanet Enterprise Server:

```
Cluster SUNWCreq add

Package SUNWntpu add

Package SUNWntpr add

Package SUNWlibC add
```

To use Network Time Protocol for server clock synchronization, you must install the two Solaris Operating Environment `ntp` packages: `SUNWntpu` and `SUNWntpr`.

The `SUNWlibC` contains the critical shared library `libC.so` that is required by the iPlanet installation routines. If this library is not present, the installation routine fails with a linker error.

The `bp-iplanetes.profile` specifies both the Solaris Operating Environment install cluster and any additional packages that will be installed by Jumpstart software.

3. Install all Patches

Before making any other changes to the system, it is critical that all recommended, security, and software vendor recommended patches be installed. A Sun recommended patch cluster containing all recommended and security patches is available from http://sunsolve.sun.com and is sorted by OS. Access to these patch clusters does not require a service contract.

It is important to install patches before any other changes are made to the server. This is especially true when the goal is to minimize the number of installed packages, since patches often install packages which may not be necessary. Be sure to always install the Recommended and Security Patch cluster at this point in the process.

Both the Recommended Patches and Patches Containing Security Fixes cluster available from http://sunsolve.sun.com include the latest Kernel Update Patch. For Solaris 7 Operating Environment, this is currently patch ID 106541. The `pkgmap` of this patch, tells us that the following files, will be updated when the patch is installed:

- /etc/rc2.d/S71rpc
- /etc/syslog.conf
- /etc/init.d/rpc

The presence of any of these files may either enable a service that has previously been disabled (rpc, automounter, volume manager) or overwrite a file with specific configuration information in it (syslog.conf). At this stage of the methodology, this is not critical as no packages have been removed, nor have any configuration changes been made to the system. Once package removal and system configuration has begun, patch installation should only be done after the index of package has been reviewed for possible conflicts.

4. Remove all Unnecessary Packages

Once the Solaris Operating Environment has been installed and patched, unneeded packages should be removed. The package removal process deletes all packages not explicitly required by either the OS or the software package being installed. On our `sun4m/SPARC/SBUS/headless` environment lab we were able to remove 29 of the 39 packages included in the Solaris Operating Environment *Core Cluster*. This may not be appropriate for all installations. Different hardware architectures, environments, and software packages may require other packages.

This package removal was automated with the `bp-iplanetes-pkg-rm.fin` script. The script is both application and OS specific, as each software package and OS have slightly different requirements.

Additional configuration and hardening of the Operating System will not be covered in this article. Look to future articles for methodologies, processes, procedures, automated scripts, and procedures on how to perform those tasks.

5. Use JumpStart Software to Configure the OS for the DataCenter Environment

Due to the repetitive nature of the installations required in this methodology, the basic network configuration steps for a server have been automated. This includes both required network and operating systems configuration. These specific scripts were customized for the test environment and not included with this document.

6. Install and Configure the Software Package

The final step in the automated portion of the methodology is to install and configure as much of the software package as possible. In the case of the iPlanet software, the only task is to extract the source packages in an appropriate directory. Once extracted, the installation routines must be run manually to configure the server. The `bp-iplanetes.fin` script mounts the appropriate directory from the Jumpstart server and extracts the package into the `/opt` directory of the server.

7. Check the Logs for Errors, if Necessary, Fix the Errors and Repeat the Installation Process

Before continuing with the installation of the software, it is important that the installation logs on the server be examined for any errors or configuration problems. The Jumpstart logs are located in `/var/sadm/system/logs` directory. The `begin.log` contains all pre-OS installation operations while the `finish.log` contains all post-OS installation steps. Usually the `finish.log` contains the interesting messages. If errors are found they must be fixed and the installation repeated. This process should be performed until all problems are resolved.

8. Test the Software Installation

At this point, the software should be installed and tested. In the case of the iPlanet Enterprise Server 4.0 software, this was a relatively straightforward task. The setup routine was run and a default configuration selected for both the administrative and production web server ports. Once configured, the `startconsole` command was used to start up the admin server. This command attempted to launch a local

Netscape session; this failed because Netscape was not installed locally on the system. Rather then managing the installation locally, a remote Netscape session was used to configure the web server through the administration port.

Overview of Final Configuration: iPlanet Enterprise Server 4.0

The process, procedures, and scripts previously defined were used to determine the minimal OS installation for iPlanet software. The minimum Solaris Operating Environment cluster was installed (*Core*). In addition, three other packages were added to the initial OS installation. The `SUNWlibC` package was added as the iPlanet installation routines required some libraries included in this package. The two NTP packages were not required by iPlanet, but were needed to address operational issues of date/time synchronization. This is particularly important for logging and auditing across many systems and will be discussed in future articles.

The following package listings were required for iPlanet Enterprise Server 4.0. Package listings are provided both for Solaris 7 and 2.6 Operating Environment. This product does not support Solaris 2.5.1 Operating Environment.

- `SUNWCreq` (cluster)
- `SUNWlibC` (package)
- `SUNWntpr` (package)
- `SUNWntpu` (package)

The package removes were specific to each version of the operating system. Details are provided for both Solaris 7 and 2.6 Operating Environment below:

Solaris 7 Operating Environment

The following packages were removed:

- SUNWsolnm
- SUNWqfed
- SUNWpsdpr
- SUNWpcser
- SUNWpcmem
- SUNWpcmcu
- SUNWpcelx
- SUNWpcmci
- SUNWos86u

- SUNWhmd
- SUNWatfsu
- SUNWatfsr
- SUNWxwdv
- SUNWdtcor
- SUNWxwmod
- SUNWcg6
- SUNWdfb
- SUNWkey
- SUNWadmr
- SUNWloc
- SUNWftpr
- SUNWftpu
- SUNWsndmr
- SUNWsndmu
- SUNWter
- SUNWploc
- SUNWploc1
- SUNWnisu
- SUNWnisr

Final Solaris 7 Operating Environment package listing for an iPlanet Enterprise Server is:

system	SUNWcar	Core Architecture, (Root)
system	SUNWcsd	Core Solaris Devices
system	SUNWcsl	Core Solaris, (Shared Libs)
system	SUNWcsr	Core Solaris, (Root)
system	SUNWcsu	Core Solaris, (Usr)
system	SUNWesu	Extended System Utilities
system	SUNWkvm	Core Architecture, (Kvm)
system	SUNWlibC	Sun Workshop Compilers Bundled libC
system	SUNWlibms	Sun WorkShop Bundled shared libm
system	SUNWntpr	NTP, (Root)
system	SUNWntpu	NTP, (User)
system	SUNWswmt	Install and Patch Utilities

The total disk space used for these twelve packages was less then 40 MBytes.

Solaris 2.6 Operating Environment

The following packages were removed:

- SUNWsolnm
- SUNWqfed
- SUNWpsdpr
- SUNWpcser

- SUNWpcmem
- SUNWpcmcu
- SUNWpcelx
- SUNWpcmci
- SUNWos86u
- SUNWhmd
- SUNWatfsu
- SUNWatfsr
- SUNWxwdv
- SUNWdtcor
- SUNWxwmod
- SUNWcg6
- SUNWdfb
- SUNWkey
- SUNWadmr
- SUNWloc
- SUNWploc
- SUNWnisu
- SUNWnisr

Final Solaris 2.6 software package listing for an iPlanet Enterprise Server is:

- system SUNWcar Core Architecture, (Root)
- system SUNWcsd Core Solaris Devices
- system SUNWcsr Core Solaris, (Root)
- system SUNWcsu Core Solaris, (Usr)
- system SUNWesu Extended System Utilities
- system SUNWkvm Core Architecture, (Kvm)
- system SUNWlibC SPARCompilers Bundled libC
- system SUNWlibms Sun WorkShop Bundled shared libm
- system SUNWntpr NTP, (Root)
- system SUNWntpu NTP, (User)
- system SUNWswmt Patch Utilities

The total disk space used by these eleven packages was approximately 40 MBytes.

References

Garfinkel, Simon and Spafford, Gene; *Practical Unix and Internet Security*, Published by O'Reilly & Associates; 04/1996; ISBN: 1565921488

Jumpstart Setup and Troubleshooting PSD/FAQ available for customers with support contracts at:

http://sunsolve.sun.com/private-cgi/retrieve.pl?doc=infodoc%2F1332&zone_32=jumpstart

Solaris Advanced System Installation (from http://docs.sun.com – query on “advanced installation configuration solaris 7”

NIPC Cyber Threat Assessment - Statement for the Record of Michael A. Vatis Director, National Infrastructure Protection Center Federal Bureau of Investigation before the Senate Judiciary Committee Subcommittee on Technology and Terrorism Washington, D.C. October 6, 1999

Computer Crime and Security Survey 1999 by Computer Security Institute (http://www.gocsi.com) and the San Francisco Federal Bureau of Investigation (FBI) Computer Intrusion Squad.

Related Web Sites

Computer Emergency Response Team (http://www.cert.org) is federally funded research and development center working with computer security issues.

Security Focus (http://www.securityfocus.org) is a web site dedicated to discussing topics of relevance to security.

The rootshell.com web site (http://www.rootshell.com) provides a searchable list of vulnerabilities posted to the various full-disclosure mailing lists.

The attrition web site http://www.attrition.org maintains an archive of defaced sites for those interested in what has happened to others.

Appendix A: Scripts

Scripts

- bp-iplanetes.profile
- bp-iplanetes.driver
- bp-iplanetes-pkg-rm.fin
- bp-iplanetes.fin

These shell scripts can be downloaded at http://www.sun.com/blueprints/tools/

Author's Bio: Alex Noordergraaf

Alexander Noordergraaf has over eight years of experience in the area of Computer and Network Security. As a Senior Security Architect for SunPS Global Enterprise Security Service (GESS), he has worked with many Fortune 500 companies on projects that include Security Assessments, Architecture Development, Architectural Reviews, and Policy/Procedure review and development. His customers have included major telecommunication firms, financial institutions, ISPs, and APSs.

Author's Bio: Keith Watson

Keith Watson has spent the past two years at Sun developing an enterprise network security auditing tool suite named the Sun Enterprise™ Network Security Service (http://www.sun.com/software/communitysource/senss/). He currently works for the SunPS Global Enterprise Security Service (GESS) consulting practice. Prior to joining Sun, he was part of the Computer Operations, Audit, and Security Technologies (COAST) laboratory at Purdue University.