Solaris System Administration Series

Solaris™ 10 Security Essentials

Sun Microsystems Security Engineers

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and the publisher was aware of a trademark claim, the designations have been printed with initial capital letters or in all capitals.

The authors and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

Sun Microsystems, Inc., has intellectual property rights relating to implementations of the technology described in this publication. In particular, and without limitation, these intellectual property rights may include one or more U.S. patents, foreign patents, or pending applications. Sun, Sun Microsystems, the Sun logo, J2ME, J2EE, Solaris, Java, Javadoc, Java Card, NetBeans, and all Sun and Java based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc., in the United States and other countries. UNIX is a registered trademark in the United States and other countries, exclusively licensed through X/Open Company, Ltd.

THIS PUBLICATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. THIS PUBLICATION COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THE PUBLICATION. SUN MICROSYSTEMS, INC., MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS PUBLICATION AT ANY TIME.

The publisher offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales, which may include electronic versions and/or custom covers and content particular to your business, training goals, marketing focus, and branding interests. For more information, please contact:

U.S. Corporate and Government Sales
(800) 382-3419
corpsales@pearsontechgroup.com

For sales outside the United States please contact:

International Sales
international@pearsoned.com

Visit us on the Web: informit.com/ph

Library of Congress Cataloging-in-Publication Data
Solaris 10 security essentials / Sun Microsystems security engineers.
p. cm.
Includes index.
ISBN 978-0-13-701233-6 (pbk. : alk. paper)
1. Solaris (Computer file) 2. Computer security 3.
Operating systems (Computers). I. Sun Microsystems.
QA76.9.A25S65524 2009
005.8—dc22
2009034603

Copyright © 2010 Sun Microsystems, Inc.
4150 Network Circle, Santa Clara, California 95054 U.S.A.
All rights reserved.

Printed in the United States of America. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. For information regarding permissions, write to:

Pearson Education, Inc.
Rights and Contracts Department
501 Boylston Street, Suite 900
Boston, MA 02116
Fax: (617) 671-3447

ISBN-13: 978-0-13-701233-6
ISBN-10: 0-13-701233-0
Text printed in the United States on recycled paper at RR Donnelley in Crawfordsville, Indiana.
First printing, November 2009

Preface

Solaris™ 10 Security Essentials

Solaris™ 10 Security Essentials is the first book in the new series on Solaris system administration. It covers all of the features of the Solaris 10 Operating System that make it the best choice for meeting the present-day challenges to robust and secure computing. Other books in the series are *Solaris™ 10 System Administration Essentials* and *Solaris™ 10 ZFS Essentials*. The former covers all of the breakthrough features of the Solaris 10 Operating System in one place. *Solaris™ 10 ZFS Essentials* provides a hands-on look at the revolutionary new ZFS file system introduced in the Solaris 10 OS.

The Solaris OS has a long history of innovation. The Solaris 10 OS is a watershed release that includes features such as:

- **Zones,** which provide application isolation and facilitate server consolidation
- **ZFS,** the file system that provides a new approach to managing your data with an easy administration interface
- The **Fault Management Architecture,** which automates fault detection and resolution
- The **Service Management Facility,** a unified model for services and service management on every Solaris system
- **Dynamic Tracing (DTrace),** for troubleshooting OS and application problems on production systems in real time

Security has long been a core strength of the Solaris OS and it has been significantly enhanced in the Solaris 10 version in areas such as:

- Zones virtualization security
- System hardening
- Trusted Extensions
- Privileges and Role-Based Access Control (RBAC)
- Cryptographic services and key management
- Auditing
- Network security
- Pluggable Authentication Modules (PAM)

The strength of Solaris operating system security is its scalability and adaptability. It can protect a single-user system with login authentication to Internet and intranet configurations.

This book is the work of the engineers and architects who conceptualized the services, wrote the specifications, and coded the Solaris OS's security features. They bring a wide range of industry and academic experience to the business of creating and deploying secure operating systems. These are the people who know Solaris 10 security best. They have combined to write a book that speaks to readers who want to learn Solaris or who want to use Solaris for the first time in their company's or their own environment. Readers do not have to be experienced Solaris users or operating system developers to take advantage of this book.

Books in the Solaris System Administration Series

Solaris™ 10 System Administration Essentials

Solaris™ 10 System Administration Essentials covers all of the breakthrough features of the Solaris 10 Operating System in one place. It does so in a straightforward way that makes an enterprise-level operating system accessible to system administrators at all levels.

Solaris™ 10 System Administration Essentials provides a comprehensive overview along with hands-on examples of the key features that have made Solaris the leading UNIX operating system for years and the significant new features of Solaris 10 that put it far ahead of its competitors. These features include Zones, the ZFS file system, Fault Management Architecture, Service Management Facility, and DTrace, the dynamic tracing tool for troubleshooting OS and application problems on production systems in real time.

Solaris™ 10 ZFS Essentials

Solaris™ 10 ZFS Essentials presents the revolutionary Zettabyte File System introduced in Solaris 10. It is a file system that is elegant in its simplicity and the ease with which it allows system administrators to manage data and storage.

ZFS is an all-purpose file system that is built on top of a pool of storage devices. File systems that are created from a storage pool share space with the other file systems in the pool. Administrators do not have to allocate storage space based on the intended size of a file system because file systems grow automatically within the space that is allocated to the storage pool. When new storage devices are added, all file systems in the pool can immediately use the additional space.

Intended Audience

The books in the Solaris System Administration Series can benefit anyone who wants to learn more about the Solaris 10 operating system. They are written to be particularly accessible to system administrators who are new to Solaris, and people who are perhaps already serving as administrators in companies running Linux, Windows, and/or other UNIX systems.

If you are not presently a practicing system administrator but want to become one, then this series, starting with *Solaris™ 10 System Administration Essentials*, provides an excellent introduction. In fact, most of the examples used in the books are suited to or can be adapted to small learning environments like a home setup. Even before you venture into corporate system administration or deploy Solaris 10 in your existing IT installation, these books will help you experiment in a small test environment.

OpenSolaris

In June 2005, Sun Microsystems introduced OpenSolaris, a fully functional Solaris operating system release built from open source. While the books in this series focus on Solaris 10, they often incorporate aspects of OpenSolaris. Now that Solaris has been open-sourced, its evolution has accelerated even beyond its normally rapid pace. The authors of this series have often found it interesting to introduce features or nuances that are new in OpenSolaris. At the same, many of the enhancements introduced into OpenSolaris are finding their way into Solaris 10. Whether you are learning Solaris 10 or already have an eye on OpenSolaris, the books in this series are for you.

3

System Protection with SMF

All services on a Solaris 10 system are controlled by the Service Management Facility (SMF). Among the advantages of SMF, which include automatic starting of dependent services and the ability to recover easily from a service outage, is the ability to use the power of role-based access control (RBAC) in an SMF manifest. With RBAC, programs can run with the precise privileges and authorizations that the program needs, and no more. This chapter shows you how to configure four programs—NFS, IP filter, FTP, and the Apache2 Web server*—as SMF services.*

3.1 Service Management Facility (SMF)

SMF provides a more powerful administrative interface for Solaris services than the traditional UNIX run-control scripts.

Solaris services are executables such as system processes, daemons, applications, and scripts. Database software, Web server software, and site-specific scripts can be controlled by SMF. SMF provides simple, fast, and visible administration through the following features.

- Services can be enabled, disabled, or restarted with one administrative command, `svcadm`.
- Failed services are restarted automatically in dependency order. The source of the failure does not affect the automatic restart.

- Service objects can be viewed and managed with commands such as `svcs`, `svcadm`, and `svccfg`.
- Services are easy to debug. The `svcs -x` command provides an explanation of why a service is not running. Per-service log files also simplify debugging.
- Services are easy to test, back up, and restore to a particular configuration because configuration states are preserved in service manifests.
- Systems boot and shut down faster because services are started and stopped according to the dependencies between services. Services can be started in parallel.
- Administrators can securely delegate tasks to non-root users who have permissions to administer particular services through RBAC rights profiles, roles, authorizations, or privileges.
- SMF *milestones* correspond to system init states such as the multiuser run level.
- SMF can be used on a system that is also using traditional UNIX `rc` scripts. While this practice is not recommended, you can use traditional scripts for some services and use SMF for others. For more information, see the `smf`(5), `svcadm`(1M), `svcs`(1), and `svccfg`(1M) man pages.

Manifests, or snapshots of each service, are in a central repository. This overall snapshot initializes the system at reboot. You can define a number of manifest collections, which are called *profiles*. The limited profile was discussed in Chapter 2, “Hardening Solaris Systems.” The `svccfg apply` *`profile`* command configures your system with *profile*.

3.2 How SMF Configuration Works

A service is shipped together with an SMF manifest. The manifest’s format is an XML file in the */var/svc/manifest/* directory. The manifest contains the information about dependencies, if the service is enabled or disabled, and other basic configuration and default information. During system boot, the manifests are imported into the SMF repository. The repository is a database in the */etc/svc/* directory.

You can have multiple manifests or snapshots of each service. At boot, a profile is selected. A profile enables or disables every Solaris service. After the profile initializes the system during boot, an administrator can further customize the configuration by using SMF commands. These commands directly modify the repository and the profile, and the changed configuration is restored at the next boot.

3.3 Modifying Solaris Services Defaults

On a Solaris system that is hardened by the limited profile, network services that you might want to run on particular systems are disabled (hardening is discussed in Chapter 2, “Hardening Solaris Systems”). For example, the `ftp` service is disabled, as is NFS file sharing. Services that require configuration, such as IPfilter and IPsec, are disabled by default.

The following sections provide examples of using SMF to configure a system for a particular purpose. Once you have configured the system, the manifest is in the repository. When the system reboots, that configuration is restored. The examples illustrate the following points.

- Services that must be configured in configuration files are enabled after the files are configured. If you did not configure the file, or if the file cannot be read, the problem is recorded in the log file.
- You might want to try different configurations of a service. By using different configuration files, you can create testing environments. The final configuration state is restored at reboot.
- Some services, such as FTP, are necessary but might require monitoring. You can set up monitoring services before bringing them online, thereby ensuring that the service is in compliance with site security policy for its first use.
- You might want to limit the attack surface on a network service. The Apache2 Web service can be configured to use RBAC to limit the privileges that the service uses. You can also require a more limited account than `root` to run the service.

3.3.1 Configuring the NFS Service

To configure a service that requires you to customize a configuration file, you perform the following steps.

1. List the status of the service.
2. Modify or create the configuration file.
3. Enable the service.
4. Verify that the service is online.
5. If the system reports an error, read the service log and then fix the error.
6. Test and use the service.

In the following example, you configure a system to serve help documents. The files must be shared read-only.

```
# svcs -a | grep nfs
...
disabled        Jan_10 svc:/network/nfs/server:default
# vi /etc/dfs/dfstab
...
share -F nfs -o ro /export/helpdocs
...
# svcadm enable svc:/network/nfs/server
# svcs -x svc:/network/nfs/server:default
State: online since Tue Jan 20 5:15:05 2009
  See: nfsd(1M)
  See: /var/svc/log/network-nfs-server:default.log
Impact: None
```

If you try to enable a service without its supporting files, view the log file to determine the problem:

```
# svcs -x svc:/network/nfs/server:default (NFS server)
 State: disabled since Tue Jan 20 5:10:10 2009
Reason: Temporarily disabled by an administrator.
   See: http://sun.com/msg/SMF-8000-1S
   See: nfsd(1M)
   See: /var/svc/log/network-nfs-server:default.log
Impact: This service is not running.
# vi /var/svc/log/network-nfs-server:default.log
...
No NFS filesystems are shared
...
```

3.3.2 Configuring the IP Filter Service

Like the NFS service, the IP filter service cannot be enabled until you create a configuration file. Your site's security requirements dictate what configuration rules you place in the file. Some services, such as IPsec, require that each communicating system has a configuration file. To enable a service that requires a configuration file involves the following steps.

1. Create the configuration file. Use the man page for the service name if you do not know the name of the configuration file. Then read the configuration file man page for the syntax.
2. If syntax verification is available, verify the syntax of the file.
3. If the service needs to run on both systems, such as the IPsec service, configure the second system.
4. Enable the service on one or both systems.

5. Enable the service.
6. Verify that the service is running.

In the following examples, you protect a system that includes non-global zones. The IP filter rules protect the global zone and the Web server zone. You first create and add rules to the `/etc/ipf/ipf.conf` configuration file.

```
# vi /etc/ipf/ipf.conf
set intercept_loopback true;
# *** GLOBAL ZONE: (IN: TCP/22, OUT: ANYTHING)
pass in quick proto tcp from any to global-zone port = 22
keep state keep frags
pass out quick from global-zone to any keep state keep frags
# *** Web Server ZONE: (IN: TCP/80, OUT: NOTHING)
pass in quick proto tcp from any to websvc port = 80
keep state keep frags
block out quick from websvc to any

# *** DEFAULT DENY
block in log all
block in from any to 255.255.255.255
block in from any to 127.0.0.1/32
```

Then you verify the syntax of the configuration file before enabling the service.

```
# ipf /etc/ipf/ipf.conf
# svcs -a | grep ipf
disabled       Dec_10   svc:/network/ipfilter:default
# svcadm enable svc:/network/ipfilter:default
# svcs ipfilter
enabled       Jan_10   svc:/network/ipfilter:default
```

To test a different configuration, you create another configuration file, verify the syntax of the file, and change the `config/entities` property to point to the new file. This test file adds rules for the Web data zone.

```
# vi /etc/ipf/testipf.conf
set intercept_loopback true;
# *** GLOBAL ZONE: (IN: TCP/22, OUT: ANYTHING)
pass in quick proto tcp from any to global-zone port = 22
keep state keep frags
pass out quick from global-zone to any keep state keep frags

# *** Web Server ZONE: (IN: TCP/80, OUT: NOTHING)
pass in quick proto tcp from any to websvc port = 80
keep state keep frags
```

continues

```
block out quick from websvc to any
# *** Web Data ZONE: (IN: TCP/22, OUT: ANYTHING)
pass in quick proto tcp from any to webdat port = 22
keep state keep frags
pass out quick from webdat to any keep state keep frags
# *** DEFAULT DENY
block in log all
block in from any to 255.255.255.255
block in from any to 127.0.0.1/32
# ipf /etc/ipf/testipf.conf
# svcprop ipfilter | grep config
config/entities fmri file://localhost/etc/ipf/ipf.conf
config/grouping astring require_all
config/restart_on astring restart
config/type astring path
# svccfg -s /network/ipfilter \
setprop config/entities=file://localhost/etc/ipf/testipf.conf
```

After you refresh and restart the service, you then verify that the property has been set.

```
# svcadm refresh ipfilter
# svcadm restart ipfilter
# svcprop ipfilter | grep etc
config/entities fmri file://localhost/etc/ipf/testipf.conf
```

After testing is complete, you can restore the original file.

```
# svccfg -s /network/ipfilter \
setprop config/entities=file://localhost/etc/ipf/ipf.conf
# svcadm refresh ipfilter
# svcadm restart ipfilter
# svcprop ipfilter | grep etc
config/entities fmri file://localhost/etc/ipf/ipf.conf
```

3.3.3 Configuring the `ftp` Service

The `ftp` service is controlled by the `inetd` command. Often, site security policy requires that an FTP server log detailed records of all FTP connections. In the following two examples, you configure properties of the `ftp` service that log transactions and turn on debugging.

To configure a service that requires you to change service properties, you perform the following steps.

1. List the status of the service.
2. List the properties of the service.
3. Change one or more properties of the service.

4. Verify that the service property is changed.
5. Enable the service.
6. Verify that the property change is effective.

In the first part of this example, you configure FTP to log every login on System A, the FTP server. Note that the `ftp` service is initially disabled on System A.

```
A # svcs ftp
STATE          STIME    FMRI
disabled       Jan_10   svc:/network/ftp:default
A # inetadm -l svc:/network/ftp:default
SCOPE    NAME=VALUE
         name="ftp"
         endpoint_type="stream"
         proto="tcp6"
         isrpc=FALSE
         wait=FALSE
         exec="/usr/sbin/in.ftpd -a"
         user="root"
...
default  tcp_trace=FALSE
default  tcp_wrappers=FALSE
default  connection_backlog=10
```

The login log property for the `ftp` service is `tcp_trace`. You change the value from `FALSE` to `TRUE`, then enable the service and verify that the service is online.

```
A # inetadm -m svc:/network/ftp:default tcp_trace=TRUE
A # inetadm -l svc:/network/ftp:default
SCOPE    NAME=VALUE
         name="ftp"
...
         tcp_trace=TRUE
...
A # svcadm enable svc:/network/ftp:default
A # svcs ftp
STATE          STIME    FMRI
online         07:07:07 svc:/network/ftp:default
```

Then, as a regular user, run the `ftp` command from machine B.

```
B $ ftp A
Connected to A.
220 A FTP server ready.
Name (A:testftp):
331 Password required for testftp.
Password:
230 User testftp logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

As superuser, examine the login record in the log file on machine A.

```
A # tail -1 /var/adm/messages
Jan 10 07:20:20 A inetd[16208]: [ID 317013 daemon.notice] ftp[6036] from B 49822
```

To continue with this example, disable the service. You want to establish monitoring before the service is online.

```
A # svcadm disable ftp
A # svcs -x ftp
svc:/network/ftp:default (FTP server)
 State: disabled since January 20, 2009  07:20:22 AM PST
Reason: Disabled by an administrator.
   See: http://sun.com/msg/SMF-8000-05
   See: in.ftpd(1M)
   See: ftpd(1M)
Impact: This service is not running.
```

The `exec` property for the `ftp` service contains the command that is executed to start the service. The man page for that command describes the arguments that the command accepts. You can select arguments to add to the `exec` property so that the command runs with those arguments when the service starts. Therefore, to modify the command that runs a service, you perform the following steps.

1. List the `exec` property of the service.
2. From the man page, determine the arguments to the service's `exec` command.
3. Add selected arguments to the `exec` property of the service.
4. Verify that the `exec` property is changed.
5. Enable the service.
6. Test and use the service.

In the following example, you modify the `ftp` service to provide debugging information and a detailed log of each transaction. To modify the `exec` property of the `ftp` service, first list the `exec` property, then read the man page of the `exec` command to determine which arguments to pass to the command.

```
# inetadm -l svc:/network/ftp:default | grep exec
         exec="/usr/sbin/in.ftpd -a"
# man in.ftpd
```

From the in.ftpd(1M) man page, select the options that provide detailed information.

- -v Write debugging information to syslogd(1M).
- -w Record each user login and logout in the wtmpx(4) file.
- -i Log the names of all files received by the FTP Server to xferlog(4).

Modify the exec property for the service and verify that the property is changed.

```
# inetadm -m ftp exec="/usr/sbin/in.ftpd -a -i -v -w"
# inetadm -l ftp | grep exec
       exec="/usr/sbin/in.ftpd -a -i -v -w"
```

Test that the property change is effective. First, enable the service. Then, as a regular user, transfer a file. Finally, verify that the log file was updated.

```
A # svcadm enable svc:/network/ftp:default
A # svcs ftp
STATE          STIME    FMRI
online         07:07:07 svc:/network/ftp:default
```

As a regular user, try to put a file in the FTP repository.

```
B $ ftp A
Connected to A.
220 A FTP server ready.
Name (A:testftp):
331 Password required for testftp.
Password:
230 User testftp logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> mput design.tar
mput design.tar? y
200 PORT command successful.
150 Opening BINARY mode data connection for design.tar.
226 Transfer complete.
^D
ftp> 221-You have transferred 0 bytes in 0 files.
221-Total traffic for this session was 361 bytes in 0 transfers.
221-Thank you for using the FTP service on A.
221 Goodbye.
B $
```

As superuser, examine the record in the xferlog file. The log indicates that the user's attempt to transfer the design.tar file from B to A was unsuccessful.

```
A # cat /var/log/xferlog
Sat Jan 20 07:18:10 2009 1 B.mydomain.com 0 /home/test/design.tar b _ i r test ftp 0 * c
```

3.3.4 Configuring the Apache2 Web Service

The Apache2 Web server program is offered as part of the Solaris OS. Web servers are frequently the targets of attackers. You can use RBAC to limit the server's vulnerability to attack. Other Solaris features, such as zones, are also useful when setting up network services.

To configure a service with RBAC, you perform some of the following steps.

1. List the properties of the service.
2. Create a rights profile, or a role, or authorizations for the service.
3. Add privileges to or remove privileges from the service.
4. Verify that the service properties are changed.
5. Enable the service.
6. Verify that the property change is effective.

The Apache2 Web server program is provided in the SUNWapch2r and SUNWapch2u packages. By default, the Apache2 service is disabled.

```
# svcs apache2
disabled 11:11:10 svc:/network/http:apache2
```

By default, services are started with the `root` account. However, the `http.conf` file for the Apache2 service creates a daemon, `webservd`, to run the service. When you configure the service with the default files, the service starts under the `root` account, switches to the `webservd` account, and runs with all privileges.

To reduce the privileges of the Apache2 server and start the service with `webservd`, do the following.

- Remove basic privileges that the service does not need, `proc_session`, `proc_info`, and `file_link_any`.
- Add the network privilege the service needs to use a privileged port, `net_privaddr`.
- Do not change the limit set.
- Set the user and group to `webservd`. When the user and group are set in the SMF manifest, the service starts as `webservd`, not as `root`.

```
# svccfg -s apache2
... apache2> setprop start/user = astring: webservd
... apache2> setprop start/group = astring: webservd
... apache2> setprop start/privileges = astring:
basic,!proc_session,!proc_info,!file_link_any,net_privaddr
... apache2> end
# svcadm -v refresh apache2
Action refresh set for svc:/network/http:apache2.
```

To verify that this configuration has been set, examine the service's start properties.

```
# svcprop -v -p start apache2
start/exec astring /lib/svc/method/http-apache2\ start
...
start/user astring webservd
start/group astring webservd
start/privileges astring
basic,!proc_session,!proc_info,!file_link_any,net_privaddr
start/limit_privileges astring :default
start/use_profile boolean false
...
```

Note

If you had created a rights profile that included the privileges for the service, you could type the name of the rights profile as the value for the `use_profile` property, rather than setting the privileges.

You can now enable the service. Verify that the service starts under `webservd` and has a limited set of privileges.

```
# svcadm -v enable -s apache2
svc:/network/http:apache2 enabled.
# svcs apache2
STATE STIME FMRI
online 12:02:21 svc:/network/http:apache2
# ps -aef | grep apache | grep -v grep
webservd 5568 5559 0 12:02:22 ? 0:00 /usr/apache2/bin/httpd -k start
...
# ppriv -S 5559
5559: /usr/apache2/bin/httpd -k start
flags = <none>
E: net_privaddr,proc_exec,proc_fork
I: net_privaddr,proc_exec,proc_fork
P: net_privaddr,proc_exec,proc_fork
L: limit
```

For more examples of using RBAC in SMF manifests, see Chapter 5, "Privileges and Role-Based Access Control."

Further Reading

For a fuller account of setting up an Apache2 Web server, see the following:

Limiting Service Privileges in the Solaris™ 10 Operating System,
`http://www.sun.com/blueprints/0505/819-2680.pdf`

Understanding the Security Capabilities of Solaris Zones Software,
`http://www.sun.com/offers/details/820-7017.html`

Eliminating Web Page Hijacking Using Solaris 10 Security,
`http://www.sun.com/software/solaris/howtoguides/s10securityhowto.pdf`

Index

A

ABI (Application Binary Interface), 222
absolute modes, 41–42
Access Control Lists (ACLs)
  access permissions in, 48–50
  getfacl for, 46–47
  inheritance flags and, 50
  setfacl for, 46–47
  in ZFS, 48–51
account lockouts, 19
accountability, 146
Active Directory (AD), 217–218
administration
  of Cryptographic Framework, 122–125
  of Kerberos, 207–208
  of Key Management Framework, 134–139
  ZFS delegated, 59–61
  of zones virtualization security, 226–229
AH (Authentication Header), 180–181
ALF (Audit Log Fetcher), 164–165
anti-spoofing, 176
Apache Web servers
  Cryptographic Framework and, 130
  Service Management Facility with, 38–39
  user-level commands for, 122
  for zones virtualization security, 227–229
Application Binary Interface (ABI), 222
applications
  Kerberos and, 215
  PKCS #11 and, 118
  third-party, 126–127
  in zones virtualization, 222–223
architecture in zones virtualization, 222–223
AS (Authentication Service), 203
attack surfaces, 230
audit classes, 167–168
Audit Log Fetcher (ALF), 164–165
audit-review roles, 165–166
audit trails. *See also* auditing
  analyzing, generally, 157
  AUE_prof_cmd records in, 162–163
  compression of, 165
  examination of, 160–163
  IDs, audit vs. real or effective, 161
  introduction to, 147
  log rotation and, 164
  managing, 163–165
  record details in, 157–159, 162–163
  remote storage of, 164–165
  use_of_privilege tokens in, 162
  user modification of files in, 162

auditing
  audit classes, creating, 167–168
  audit-review roles in, 165–166
  audit trails in. *See* audit trails
  audit user IDs in, 147, 161
  audit_class files in, 149
  audit_control files in, 150–151, 155–156
  audit_event files in, 149–150
  audit_syslog plugs-ins, 166–167
  audit_user files in, 152, 156
  audit_warn scripts in, 156–157
  configuring for, generally, 148
  customizing, 165
  enabling, 155–156
  events in non-global zones in, 236–237
  goals of, 146
  introduction to, 145
  keywords for autid_control file, 151
  legal accountability in, 146
  notes on, 168
  performance degradation in, 147
  policies for, 147–148, 152–155
  preselection masks in, 147
  principles of, 146
  quickstart guide for, 148
  tracking systems for, 146
AUE_prof_cmd records, 162–163
authentication
  adding RSA or DSA identities, 198
  agent for, 198
  key generation for, 197–198
  methods of, 195–196
  PAM and. *See* Pluggable Authentication Modules (PAM)
Authentication Header (AH), 180–181
Authentication Service (AS), 203
authorization
  in Kerberos, 216–217
  in Printer Management, 79–80
  in role-based access control, 75–77
  in Service Management Facility and, 90–93
autoboot command, 227, 231
automatic credential renewal, 214
availability, 226

B

Basic Audit and Reporting Tool (BART)
  Create Mode, 20–21
  filesystem object attributes, collecting, 20–21
  filesystem object attributes, comparing, 21–22
  filesystem objects, genuineness of, 22
  introduction to, 20
  manifests, creating, 54–55
  manifests, current vs. stored, 55–56
  rules files for, 53–58
  Solaris Fingerprint Database and, 56
basic sets, 68
bit-fields, 42–43
bit masks, 45–46
bracketing, of privileges, 69–71
brand command, 227, 231
branded zones, 223–226
brandz, 223, 237

C

CAPP (Controlled Access Protection Profile), 145
CAs (Certificate Authorities), 189
Certificate Authorities (CAs), 189
certificate revocation lists (CRLs), 135, 189
certificate signing requests (CSRs), 57–59, 137–138
chgrp, 45
chown, 45
chroot mechanism, 222, 233–234
CIPSO template, 243–247
clients, 197, 211–215
configuration
  for auditing, 148
  of IP Filter, 170–171
  of IPSec, 181–182
  of Pluggable Authentication Modules, 101–106
  of Service Management Facility, 30
  of SunSSH, 194–195
  of Trusted Extensions, 243–244
configuration hardening
  default values in, 20

introduction to, 16–17
lock out accounts in, 19
log core dumps in, 18–19
non-executable stacks in, 18
OpenBoot PROM in, 17–18
publications on, 20
containers. *See* zones
control manifests, 55–56
Controlled Access Protection Profile (CAPP), 145
CRLs (certificate revocation lists), 135, 189
Cryptographic Framework. *See also* PKCS (Public Key Cryptography Standard) #11
administration of, 122–125
Apache Web server with, 122, 131
`cryptoadmin enable` and `disable` subcommands, 124–125
`cryptoadmin install, uninstall, load,` and `unload`, 125
`cryptoadmin list` subcommand, 122–124
`decrypt` command in, 119–120
`digest` command in, 119
drivers with, 126
`elfsign` command in, 120–121
`encrypt` command in, 119–120
hardware acceleration and, 125–127
hardware in, 128–129
introduction to, 113
Java using, 130–131
listing mechanisms, 124
OpenSSL libraries in, 121–122
PKCS #11 in, 114–119
listing providers, 122–123
using providers, 126
third-party applications using PKCS #11 API, directly, 127
third-party applications using PKCS #11 API, through NSS libraries, 127
third-party applications using PKCS #11 API, through Open SSL, 126–127
tokens, listing, 123–124
troubleshooting, 127–128
user-level commands, 119–122
using through NSS, 127, 129–130
CSRs (certificate signing requests), 57–59, 137–138

D

daemon
Internet Key Exchange, 184–185
kadmind, 207–208
Remote Shell, 107–108
restricting privileges for, 69
SunSSH, 196–197
UID as, 65
debugging PAM stacks, 111–112
decay of security, 52
decryption, 119–120
denial of service (DoS) attacks, 3, 226
Destination Network Address Translation (DNAT), 177
devices
PKCS #11 hardware, 191–192
Trusted Extensions accessing, 258–259
in zones virtualization, 225–226
`digest` command, 119
DNAT (Destination Network Address Translation), 177
Domain Name Service (DNS)
Kerberos and, 205, 212
Service Management Facility and, 90–91
DoS (denial of service) attacks, 3, 226
drivers, 126
drop boxes, storing audit logs in, 164

E

Effective set (E) privileges, 69–71
effective vs. real IDs, 161
ELF (Executable and Linking Format) objects. *See* signed ELF (Executable and Linking Format) objects
`elfsign` command, 120–121
Encapsulating Security Payload (ESP), 180–181
encapsulation, 229–230
encryption
Cryptographic Framework and. *See* Cryptographic Framework
of Network File System, 217
public key. *See* PKCS (Public Key Cryptography Standard) #11
user-level commands for, 119–120
via hardware, 128–129
escalation prevention, 71

ESP (Encapsulating Security Payload), 180–181
events, 236–237
exclusive IP stack instances, 235–236
exclusive-IP zones, 224–225
Executable and Linking Format (ELF) objects. *See* signed ELF (Executable and Linking Format) objects
execution attributes, 74–75
extended attributes, 47

F

Fault Management Resource Identifier (FMRI), 11
file-system security
  integrity of file-system in, 52
  NFSv4 ACLs for, 48–51
  NFSv4 mount options in, 58
  Solaris Fingerprint Database for, 52–54
  ZFS delegated administration, 59–61
  ZFS mount options in, 58–59
File modes, 41–42
File Transfer Protocol (FTP), 34–38, 178
file transfers with SunSSH, 199
filesystems
  collecting object attributes in, 20–21
  comparing object attributes in, 21–22
  genuineness of objects in, 22
  in Trusted Extensions, 257–258
fine-grained privileges
  bracketing, 70–71
  escalation prevention in, 71
  overview of, 66–69
  privilege sets, 69–70
  restricting process privileges, using, 71–72
Fingerprint Database
  Basic Audit and Reporting Tool in, 53–54, 56
  command-line tool in, 53
  for file-system security generally, 52–53
  MD5 file fingerprints, 25–26
  overview of, 23–25
first zones, 252–254
FMRI (Fault Management Resource Identifier), 11
`fs` command, 227
FTP (File Transfer Protocol), 34–38, 178

G

Generic Security Service Application Program Interface (GSS-API), 194
`getfacl`, 46–47
gkadmin (1M) GUI, 209
global zones, 236–237
group mode
  introduction to, 41–42
  owners of, 45
GSS-API (Generic Security Service Application Program Interface), 194

H

hardening Solaris systems
  Basic Audit and Reporting Tool in, 20–22
  configurations in. *See* configuration hardening
  Fingerprint Database in, 23–26
  introduction to, 9
  profiles in, 14–16
  references on, 26–27
  securing network services, generally, 9–10
  Service Management Facility for, 11–14
  signed ELF objects in, 22–23
  software, minimizing installed, 10–11
  Web-facing Web servers in, 232
Hashed Method Authentication Code (HMAC), 115
Health Insurance Portability and Accountability Act (HIPPA), 145
hijacking, of web pages, 232–233
HIPPA (Health Insurance Portability and Accountability Act), 145
history of Solaris Security, 1–3
HMAC (Hashed Method Authentication Code), 115
host-based intrusion detection, 236
hosts
  adding to known networks, 245–246
  in event monitoring, 236
  principal's keys and, 218–219
  remote, 244–245
  templates assigned to, 246
  trusted, 243
  on trusted networks, 246–247

I

ICMP (Internet Control Message Protocol) ECHO, 172
identities of zones, 225
IDs (identifications)
  auditing, 147
  real vs. effective, 161
  in UNIX, 65–66
IKE (Internet Key Exchange). *See* Internet Key Exchange (IKE)
incremental propagation, 209–211
`inetd` (1M), 10, 14–15
Inheritable set (I) privileges, 69–71
`inherit-pkg-dir` command, 227, 231
`in.rshd`, 107–108
inter-zone packets, 174–175
Internet Control Message Protocol (ICMP) ECHO, 172
Internet Key Exchange (IKE), 186–189
  for automated key management, 183
  certificates, 186–189
  configuration, viewing and debugging of, 185–186
  daemon configuration in, 184–185
  in IPSec configuration, 181–182
  setups, 184
introspection, 221, 236
intrusion detection, 230, 236
  host-based, 236
  network-based, 236
IP Filter
  anti-spoofing and, 176
  configuring, 170–171
  Destination Network Address Translation with, 177
  for inter-zone packets, 174–175
  introduction to, 169–170
  last vs. first matches in, 170–171
  logging with, 178–179
  loop-back packet filtering, 174–175
  Network Address Translation with, 176–178
  Service Management Facility and, 32–34
  Source Network Address Translation with, 177
  starting, 171
  stateful vs. stateless filtering with, 171–173
  troubleshooting, 173–174
  in zones, 174
IPSec (IP Security)
  configuration of, 181
  IKE, certificates for, 186–189
  IKE, daemon configuration, 184–185
  IKE for automated key management in, 183
  IKE setups, 184
  IKE, viewing and debugging configuration of, 185–186
  interoperability of, 192
  introduction to, 179
  keys, assigning manually, 183
  NAT traversals in, 191
  per-socket policies in, 192
  PKCS #11 hardware devices and, 191–192
  policy, assigning, 181–182
  technical notes on, 192
  tunnels, 189–191
isolation using zones, 229–230

J

Java
  authorizations in, 75
  Desktop System, 198
  using Cryptographic Framework, 130–131
  VM Sandbox of, 229

K

`kadm5.keytab` files, 208–209
`kadmind` daemon, 207–208
`kclient`, 212–213
KDC (Key Distribution Center). *See* Key Distribution Center (KDC)
Kerberos
  administration of, 207–208
  application servers for, generally, 215
  authorization in, 216–217
  automatic credential renewal in, 214
  clients, configuring, 211–215
  clients, zero-configuration, 212
  databases for, creating, 206
  encrypted NFS and, 217

Kerberos *(Continued)*
gkadmin (1M) GUI in, 209
incremental propagation in, 209–211
insecure networks and, 202–203
introduction to, 201–202
kadm5.keytab files in, 208–209
kadmind daemon, configuring, 207–208
kclient, 212–213
KDCs, configuring, 205–207
keytabs in, 208–209, 218–219
kinit, 213
LDAP and, 206–207
Microsoft Active Directory and, 217–219
PAM and, 214–215
passwords for, 208–209
principals in, 204
propagation, 209–211
realms in, 204
references on, 219–220
secret keys in, 202–203
slave KDCs in, 209–211
in Solaris OS, generally, 204–205
SunSSH and, 215–216
tickets in, 203
traditional propagation in, 211
Windows 2003 servers and, 218–219
Key Distribution Center (KDC)
configuring, 205–206
creating Kerberos databases with, 206–207
defined, 203
slaves, 209–211
starting, 207
Ticket Granting Service of, 203
zero-configuration clients, 212
Key Management Framework (KMF)
administrative utility, generally, 134
introduction to, 133–134
kmfcfg(1), 140–142
pktool(1), 134–139
policy configuration utility of, 140–142
policy-enforcement mechanism of, 139–140
programming interfaces of, 142–143
key operations, 134–135
keys, assigning manually, 183
keystores, 136
certificate signing requests and, 137–138
contents of, 137–138
importing PKI objects to, 135
limiting contents of, 136
symmetric keys and, 138–139
verifying contents of, 137–138
keytabs, 208–209, 218–219
kinit, 213
KMF (Key Management Framework). *See* Key Management Framework (KMF)
kmfcfg(1), 140–142

L

Labeled Zone Manager, 251–255
labeled zones, 224, 251
labeling. *See* Trusted Extensions
legal accountability, 146
Limited set (L) privileges, 69–71
limited_net profiles, 12–13
limitpriv command, 231, 234
lock out accounts, 19
log core dumps, 18–19
log rotation, 164
logging with IP Filter, 178–179
logins, 103–106
loop-back packet filtering, 174–175

M

MAC (Message Authentication Code), 55
Management Console, 77–79, 244
manifests, 30, 54–56
map rules for IP Filter, 177
masks
bit, 45–46
of classes, 149
preselection, 147, 152
mca device drivers, 126
MD5 Message Digest Algorithms
in Cryptographic Framework, 23–26
in Fingerprint Database, 52–53
Message Authentication Code (MAC), 55
message digests, 23–26
metaclusters, 11
metaslots, 116–118
Microsoft Active Directory, 217–219
milestones, in SMF, 30
MLPs (multilevel network ports), 248

MLS (multilevel security), 224
modules. *See also* Pluggable Authentication Modules (PAM)
  availability of, 98–99
  flags of, 102–106
  introduction to, 96
  stacks for logins, 97–98
  standard, 99–101
Morris worm, 1–2
mount options, 58
multilevel desktops, 255–256
multilevel network ports (MLPs), 248
multilevel security (MLS), 224

N

name-space separation, 222, 225, 230–232
ncp (Niagara cryptographic provider), 200–201
`net` command, 227, 231
Network Address Translation (NAT)
  Destination, 177
  with IP Filter, 169–170, 176–177
  proxies, 178
  Source, 177
  traversal, 191
network-based intrusion detection, 236
Network File System (NFS)
  access permissions in, 48–50
  ACLs in v4, 48–51
  encrypted, 217
  inheritance flags in, 50
  mount options in, 58
  Service Management Facility with, 31–32
  Version 4 ACLs, 48–51, 58
  ZFS for, 51
network security
  introduction to, 169
  IP Filter for. *See* IP (Internet Protocol) Filter
  IPSec for. *See* IPSec (IP Security)
  Kerberos for. *See* Kerberos
  OpenSSL, 199–201
  SunSSH. *See* SunSSH (Solaris Secure Shell)
  zones virtualization for. *See* zones virtualization security
Network Security Services (NSS), 129–130
networks
  accessing in Trusted Extensions, 259–260
  addressing. *See* Network Address Translation (NAT)
  in event monitoring, 236
  insecure, 202–203
  interfaces in Trusted Extensions, 251–252
  security of. *See* network security
  zones virtualization in, 224–225
NFS (Network File System). *See* Network File System (NFS)
Niagara cryptographic provider (ncp), 200–201
`nmap`, 16
non-executable stacks, 18
non-global zones, 225–227, 231–236
NSS (Network Security Services), 129–130

O

Object Label Management, 257
Open SSH (Secure Shell), 193–194
OpenBoot PROM (programmable read-only memory), 17–18
OpenSSL (Secure Sockets Layer)
  introduction to, 199–200
  libraries, 121–122
  PKCS #11 engines and, 200–201
operating system virtualization, 221–222, 229–235
OS virtualization, 221–222, 229–235

P

packages, 10–11, 225
packets, inter-zone, 174–175
PAM. *See* Pluggable Authentication Modules (PAM)
`pam.conf(4)` files, 101–106
passwords
  in Kerberos, 208–209
  in OpenBoot PROM, 17
  PAM and. *See* Pluggable Authentication Modules (PAM)
per-socket policies, 192
performance degradation, 147

permissions, file system, 48–50, 59–61
Permitted set (P) privileges, 69–71
PKCS (Public Key Cryptography Standard) #11
  API, 126–127
  application access to hardware providers, 118
  consumers of, 114
  hardware devices for, 191–192
  introduction to, 114
  kernel hardware providers for, 118
  kernel software providers for, 118–119
  metaslots in, 116–118
  plug-ins for, 115
  providers for, 115
  slots in, 116
  software tokens in, 116
  tokens in, 115
`pkgrm` (1M), 10
PKI (Public Key Infrastructure). *See* Public Key Infrastructure (PKI)
`pktool`(1)
  certificate signing requests with, 137–138
  introduction to, 134–135
  Key Management Framework (KMF), 134–139
  keystore contents after `gencert`, 137
  keystore contents after `gencsr`, 138
  self-signed X.509 certificates with, 136
  symmetric keys with, 138–139
plug-ins, 115
Pluggable Authentication Modules (PAM), 95–112
  adding PAM modules to, 110–111
  configuration, 101–106
  consumers, 106–109
  debugging, 111–112
  framework, 96
  introduction to, 95
  library, 109
  modules, 96–101
  Remote Shell daemon, configuration, 107–108
  SSH, configuration, 108–109
  tasks, 109–112
policies
  for auditing, 147–148, 152–155
  in IPSec, 181–182
  KMF database of, 139–142
portmap keywords, 177
`pool` command
preselection masks, 147, 152
Printer Management
  authorization in, 77–79
  normal users of, 87–88
  role assignment in, 82–86
private addresses, 177
privileges
  authorizations and, 91–93
  bracketing, 70–71
  escalation prevention in, 71
  fine-grained, generally, 66–69
  implementation details, 72
  for non-global zones, 232–235
  restricting process privileges, 71–72
  sets of, 69–70
  for system services, 90–91
  UNIX security model and, 63–66
profiles
  defined, 12–13
  in Service Management Framework, 30
  shells for, 86–87
promiscuous execution, 47
propagation, 209–211
providers
  for Cryptographic Framework, 122–123, 126
  of kernel software, 118–119
  for PKCS #11, 115
proxy commands, 199
public certificates, 188
Public Key Cryptography Standard (PKCS) #11. *See* PKCS (Public Key Cryptography Standard) #11
Public Key Infrastructure (PKI)
  importing objects to keystores, 135
  in Internet Key Exchange, 185–186, 189
  in Key Management Framework. *See* Key Management Framework (KMF)
  signing and verifying ELF objects with, 57
PUBLIC label
  in accessing networks, 258
  defined, 239, 242

in filesystem protection, 256–257
in Labeled Zone Manager, 253–254
in multilevel desktops, 255–256
template for, 245
wildcards and, 246–247

Q

quick keyword for IP Filter, 170–171

R

RBAC (role-based access control). *See* role-based access control (RBAC)
real vs. effective IDs, 161
records
audit ID, 161
in audit trails, generally, 157–159
AUE_prof_cmd, 162–163
use_of_privilege, 162
user modification of files owned by others, 162
Reduced Networking, 11
redundant services, 230–232
remote host templates, 244–245
Remote Shell daemon, 107–108
remote storage, 164–165
replicated services, 230–232
replication, 230
resource control, 226
RESTRICTED label, 239, 255
restricting processes, 71–72
rights profiles
assigning to roles, 81–86
assigning users to, 87–88
authorizations and, 91–93
implementing, 86–87
Management Console for, 77–79
overview of, 74–77
predefined, 79–81
role-based access control (RBAC)
Apache2 Web server program and, 38–39
components of, 88
definition of roles in, 73–74
introduction to, 72–73
Management Console for, 77–79
predefined rights profiles in, 79–81
privileges and. *See* privileges
right profiles in. *See* rights profiles
UNIX security model and, 63–66
using files repository, 88–90
using network repositories, 90
roles
for access control. *See* role-based access control (RBAC)
audit-review, 165–166
rights profiles assigned to, 81–86
in Trusted Extensions, 248–250, 258
rollback permissions, 61
root accounts, 63–65
root compromises, 155
rpdbind, 14–16
RSA Laboratories, 114
rules files for BART, 54

S

Sarbanes-Oxley Act (SOX), 145
SBD (Secure By Default), 2
scp command, 199
second zones, 254–255
Secure By Default (SBD), 2
Security Parameters Index (SPI), 180–181
self-signed X.509 certificates, 136
server hardening, 232–235
service instances, 11
Service Management Facility (SMF)
Apache2 Web service and, 38–39
authorization in, 90–91
configuration of, 30
controlling network services with, 11–14
defaults of Solaris services, modifying, 31
framework of, 29–30
FTP service, configuring, 34–38
introduction to, 29
IP filter service, configuring, 32–34
Network File System, configuring, 31–32
privileges in, 90–91
setfacl, 46–47
SFD. *See* Fingerprint Database
sftp command, 199
shared-IP zones, 224–225
shells, profile, 86–87
short OS virtualization. *See* zones virtualization security

signed ELF (Executable and Linking Format) objects
  introduction to, 22–23, 56
  signing and verifying, 57–58
  verifying, 23, 57
slave KDCs, 209–211
slots in PKCS #11, 116
SMF. *See* Service Management Facility (SMF)
SNAT (Source Network Address Translation), 177
soft tokens, 115
software tokens, 116
Solaris Cryptographic Framework. *See* Cryptographic Framework
Solaris Fingerprint Database (SFD). *See* Fingerprint Database
Source Network Address Translation (SNAT), 177
SOX (Sarbanes-Oxley Act), 145
SPARC
  device drivers, 126
  non-executable stacks and, 18
  OpenBoot PROM and, 17
SPI (Security Parameters Index), 180–181
`ssh`, 197
`ssh-add`, 198
`ssh-agent`, 198
`ssh-http-proxy-connect`, 199
`ssh-keygen`, 197
`ssh-socks5-proxy-connect`, 199
`sshd`, 108–109, 196–197
standard of PKCS #11. *See* PKCS (Public Key Cryptography Standard) #11
stateful vs. stateless filtering, 171–173
sticky bits, 44
storage, 55–56, 164–165
Sun Crypto Accelerator 6000, 126
Sun-wide area network (SWAN), 2–3
SunSSH (Solaris Secure Shell)
  authentication in, 195–198
  client, 197
  commands in, 196
  configuring, generally, 194–195
  daemon, 197
  file transfers, 199
  introduction to, 192–193
  Kerberos and, 215–216
  Open SSH vs., 193–194
  protocol version 2 for, 195
  proxy commands, 199
  `scp`, 199
  `sftp`, 199
  `ssh`, 197
  `ssh-add`, 198
  `ssh-agent`, 198
  `ssh-http-proxy-connect`, 199
  `ssh-keygen`, 197
  `ssh-socks5-proxy-connect`, 199
  `sshd`, 196–197
  starting and stopping, 194
  versions of, 193
superusers
  overview of, 6
  power of, 63
  in Service Management Framework, 36–38
  in zones virtualization, 222
`svcadm`, 12
SWAN (Sun-wide area network), 2–3
symbolic file modes, 41–42
symmetric keys, 138–139
`syslog`, 14–16
system protection with SMF. *See* Service Management Facility (SMF)
system services, 90–91

T

TCSEC (Trusted Computer System Evaluation Criteria), 145
`telnet` worm, 2–3
templates, 244–247
TGS (Ticket Granting Service), 203
TGT (Ticket Granting Ticket), 203, 214–215
third-party applications using PKCS #11 API
  directly, 127
  through NSS libraries, 127
  through OpenSSL, 126–127
Ticket Granting Service (TGS), 203
Ticket Granting Ticket (TGT), 203, 214–215
tokens
  in audit records, 157–159
  in Cryptographic Framework, 123–124
  in PKCS #11, 115
  `use_of_privilege`, 162
tracking systems, 146

traditional propagation, 211
trust relationships, 204
Trusted Computer System Evaluation Criteria (TCSEC), 145
Trusted Extensions
  benefits of, 239–240
  configuring, generally, 243–244
  devices, accessing, 258–259
  enabling, 240–241
  filesystem protection in, 257–258
  first zones, creating, 252–254
  hosts added to known networks, 245–246
  hosts contacted on trusted networks, limiting, 246–247
  introduction to, 239
  labeled zones, creating, 251
  labels, copying and pasting between, 257
  labels in, 242–243
  Management Console and, 244
  multilevel desktops, using, 255–256
  multilevel network ports, configuring, 248
  network interfaces, sharing, 251–252
  networks, accessing, 259–260
  processes, observing, 258
  remote host templates for, 244–245
  roles, assuming, 258
  roles, creating, 248–250
  second zones, creating, 254–255
  starting, 241–243
  summary of, 260
  templates assigned to hosts, 246
  users, creating, 250–251
  Windows, moving into other workspaces, 257
  workplace labels, changing, 256–257
  zones, creating, 252–255
Trusted Path
  defined, 239–241, 242
  labels, copying and pasting between, 242, 257
  multilevel network ports and, 248
  roles in, 248, 258
tunnels, 189–191

U

UIDs (user ID accounts)
  as daemon, 65
  privilege bracketing and, 70–71
  privilege escalation and, 71
  privileges of, 63–64
  process privileges of, 71–72
  real vs. effective, 65–66
`umask`, 45–46
UNIX file-system security
  `chgrp`, 45
  `chown`, 45
  extended attributes in, 47
  `getfacl`, 46–47
  overview of, 41–44
  promiscuous execution in, 47
  `setfacl`, 46–47
  `umask`, 45–46
UNIX security model, 63–64
unlabeled hosts, 243, 247
`use_of_privilege`, 162
user ID accounts (UIDs). *See* UIDs (user ID accounts)
user IDs. *See* IDs (identifications)
users
  audit file for, 152
  ID zero accounts, 63–64
  login service for authentication of, 103–105
  modifying files, 162
  privileges of, 162
  real vs. effective IDs of, 65–66
  in role-based access control, 88–90
  in Trusted Extensions, 250–251
  in ZFS, 59–61

V

verifying signed ELF objects, 57–58
virtual machine introspection, 221, 236
virtualization, 221–237
virtualization security, 221–237
virtual private networks (VPNs), 179
VPNs (virtual private networks), 179

W

Windows
  Microsoft, 218-219
  labeled in Trusted Extensions, 257
workplace labels, 256–257

X

X.509 certificates
- for Internet Key Exchange, 186
- operations of, 134
- policy enforcement with, 139
- self-signed, 136

Z

zero, UID accounts. *See* UIDs (user ID accounts)

ZFS
- Access Control Lists in, 48–51
- compressing files in, 165
- creating zones in, 252–255
- delegated administration, 59–61
- mount options, 58–59
- for Network File System, 51

`zlogin` command

`zoneadm` command, 223, 226, 228

`zonecfg` command, 223, 226–227, 234

`zonename` command, 227, 231, 233–236

`zonepath` command, 227, 231

zones
- exclusive-IP, 224–225
- in IP Filter, 174
- in Trusted Extensions, 252–255
- global, 224–225, 228, 230–233, 235
- non-global, 225–227, 231–236
- shared-IP, 224–225
- virtualization in. *See* zones virtualization security

zones virtualization security
- administration of, 226–229
- advantages of, generally, 229
- for Apache Web servers, 227–229
- architecture of zones in, 222–223
- branded zones in, 223–226
- CPU visibility in, 225
- devices in, 225–226
- for encapsulation, 229–230
- events in non-global zones in, 236–237
- events, monitoring, 236
- exclusive IP stack instances in, 235–236
- hardening Web-facing Web servers using, 232
- identities of zones in, 225
- introduction to, 221–222
- for isolation, 229–230
- labeled zones in, 224
- networking in, 224–225
- packaging in, 225
- privileges for non-global zones in, 232–235
- references on, 237
- replicated or redundant services using, 230–232
- resource control in, 226
- starting, 226–229