

Contents

Exposing Default Solaris Security Levels

- Consider changing the umask in /etc/profile from the default value of 022 to something more restrictive, such as 027.
- Replace insecure cleartext daemons, such as FTP, Telnet, and the Berkeley r-commands, with a secure replacement like SSH or OpenSSH.
- Create Authorized Use banners in /etc/motd and /etc/issue.

Foreword

xxi

Chapter 1 Introducing Solaris Security: Evaluating Your Risk

1

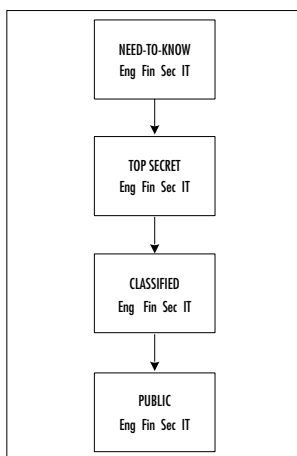
Introduction	2
Exposing Default Solaris Security Levels	2
Altering Default Permissions	2
Making Services Available after Installation	4
Using Solaris as an FTP Server	4
Using Telnet to Access a Solaris System	5
Working with Default Environmental Settings	7
Evaluating Current Solaris Security Configurations	9
Evaluating Network Services	9
Evaluating Network Processes	11
Monitoring Solaris Systems	14
Using the sdtprocess and sdtperfmeter Applications	14
Monitoring Solaris Logfiles	16
Monitoring the Access Logs	16
Monitoring the sulog	17
Validating the System Logs	17
Testing Security	18
Testing Passwords	18
Testing File Permissions	20
Securing against Physical Inspections	21
Securing OpenBoot	21
Documenting Security Procedures and Configurations	22

Documenting Security Procedures	22
Documenting System Configurations	24
Obtaining Disk Usage Information	24
Gathering System Information with vmstat	25
Summary	27
Solutions Fast Track	28
Frequently Asked Questions	30

Chapter 2 Securing Solaris with the Bundled Security Tools 33

Introduction	34
The Orange Book	35
Choosing Solaris 8 C2 Security	38
Configuring Auditing	40
Managing the Audit Log	42
Understanding Auditing Classifications	43
Configuring Auditing	44
Extracting and Analyzing Auditing Data	45
Choosing Trusted Solaris 8	47
Using Trusted Solaris 8's B1-Level Security	48
Understanding the Concept of Mandatory Access Control	50
Administrative Labels	53
Auditing and Analyzing Trusted Solaris 8	54
Solaris 8 Security Enhancements	55
Using SunScreen Secure Net	55
Utilizing SunScreen SKIP	56
Utilizing SKIP's VPN Capabilities	56
Using the Solaris Security Toolkit	58
Working with the Solaris Security Toolkit's System Files	58
Using OpenSSH	59
Summary	61
Solutions Fast Track	61
Frequently Asked Questions	63

An Example of Classification Hierarchy



Detecting Unusual Traffic with Network Traffic Monitoring

- Snoop, a built-in Solaris utility, is a powerful network tool for real-time monitoring of network activity for short periods of time.
- A dedicated sniffer/IDS system like Snort is the best way to get current and historically accurate information about network traffic types and patterns.

Chapter 3 Securing Solaris with Freeware Security Tools

67

Introduction	68
Detecting Vulnerabilities with Portscanning	71
Advanced Portscanning	76
Discovering Unauthorized Systems Using IP Scanning	77
Using the arp Command on Solaris	79
Detecting Unusual Traffic with Network Traffic Monitoring	81
Using Snoop	82
Using Snort	83
Using a Dedicated Sniffer	86
Using Sudo	88
Summary	93
Solutions Fast Track	94
Frequently Asked Questions	96

Chapter 4 Securing Your Users

99

Introduction	100
Creating Secure Group Memberships	101
Role-Based Access Control	103
Understanding Solaris User Authentication	104
Authenticating Users with NIS and NIS+	107
Authenticating Users with Kerberos	109
Authenticating Users with the Pluggable Authentication Modules	115
Summary	122
Solutions Fast Track	122
Frequently Asked Questions	125

Chapter 5 Securing Your Files

127

Introduction	128
Establishing Permissions and Ownership	129
Access Control Lists	132
Role-Based Access Control	135
/etc/user_attr -	
user:qualifier:res1:res2:attr	136

/etc/security/auth_attr - authname:res1:res2:short_desc:long_ desc:attr	137
/etc/security/prof_attr - profname:res1:res2:desc:attr	137
/etc/security/exec_attr - name:policy:type:res1:res2:id:attr	137
Changing Default Settings	138
Using NFS	142
Share and Share Alike	143
Locking Down FTP Services	145
Using Samba	147
Monitoring and Auditing File Systems	151
Summary	154
Solutions Fast Track	154
Frequently Asked Questions	156

Chapter 6 Securing Your Network 159

Watching Packets with Snoop

Here are a few examples
of when you may want to
use snoop:

- To verify that DHCP requests are being received and answered by the DHCP server
- To identify the source of denial of service (DoS) attacks
- To determine what Web sites your users are visiting
- To identify the source address of a suspected intruder
- To locate any unauthorized hosts

Introduction	160
Configuring Solaris as a DHCP Server	160
Using the dhcpmgr GUI Configuration Tool	161
Using the dhcpconfig Command-Line Tool	170
Securing DNS Services on Solaris	173
Using BIND	174
Setting Up a chroot Jail for BIND	174
Securing Zone Transfers in BIND 8	180
Configuring Solaris to Provide Anonymous FTP Services	181
Using X-Server Services Securely	182
Using Host-Based Authentication	183
Using User-Based Authentication	183
Using X-Windows Securely with SSH	186
Using Remote Commands	187
Using Built-In Remote Access Methods	187
Using SSH for Remote Access	189
Enabling Password Free Logins with	

Answers to Your Frequently Asked Questions

- Q:** What is the best way to filter traffic handled by sendmail for virii?
- A:** There are several tools available for just this purpose. Some of them are freeware and others are commercial. You should evaluate each product based on your needs and then make the choice that best suits your environment. Certain products even integrate well with certain firewalls. Sendmail itself really should not be used as a content filter—it was never designed for this purpose.

SSH	191
Summary	193
Solutions Fast Track	194
Frequently Asked Questions	195
Chapter 7 Providing Secure Web and Mail Services	199
Introduction	200
Configuring the Security Features of an Apache Web Server	201
Limiting CGI Threats	203
Using Virtual Hosts	206
Monitoring Web Page Usage and Activity	206
Configuring the Security Features of Sendmail	209
Stopping the Relay-Host Threat	213
Tracking Attachments	215
Summary	218
Solutions Fast Track	218
Frequently Asked Questions	220
Chapter 8 Configuring Solaris as a Secure Router and Firewall	223
Introduction	224
Configuring Solaris as a Secure Router	224
Reasoning and Rationale	225
Routing Conditions	225
The S30network.sh Script	226
The S69inet Script	227
Configuring for Routing	229
A Seven-Point Checklist	229
Security Optimization	233
Security Implications	233
Minimal Installation	233
Minimal Services	234
Minimal Users	235
Minimal Dynamic Information	235
Minimal Cleartext Communication	235

Steps to Ensure the System Isn't Routing Traffic

1. Check for the `/etc/notrouter` file. If it does not exist, create it.
2. Check the value of `ip_forwarding` in the IP kernel module after the system has been rebooted.
3. Test the system by attempting to reach one interface of the system through the other.

Unconfiguring Solaris Routing	236
A Three-Point Checklist	236
Routing IP Version 6	237
Configuration Files	238
The <code>hostname6.interface</code> File	238
The <code>ndpd.conf</code> File	239
The <code>ipnodes</code> File	241
The <code>nsswitch.conf</code> File	242
IPv6 Programs	242
The <code>in.ndpd</code> Program	242
The <code>in.ripngd</code> Program	243
The <code>ifconfig</code> Command	244
IPv6 Router Procedure	245
Stopping IPv6 Routing	246
Method 1: Rebooting the System	246
Method 2: Not Rebooting the System	246
IP Version 6 Hosts	247
Automatic Configuration	247
Manual Configuration	248
The <code>ipnodes</code> File	248
DNS	248
Configuring Solaris as a Secure Gateway	250
Configuring Solaris as a Firewall	250
General Firewall Theory	251
General Firewall Design	252
SunScreen Lite	253
IP Filter	254
Using NAT	254
Guarding Internet Access with Snort	255
Snort Configuration File	256
Snort Log Analysis	257
Summary	259
Solutions Fast Track	261
Frequently Asked Questions	263

Configuring Squid Services

Q: Can I force Squid to send certain requests directly to an Internet site, without using the cache? My own Web servers are local and don't need caching.

A: You can use the `dstdomain` `acl` and `always_direct` tag for this purpose:

```
acl localservers
dstdomain
.incoming-
traveller.com
always_direct
allow
localservers
```

Chapter 9 Using Squid on Solaris	265
Introduction	266
The Default Settings of a Squid Installation	266
Configuring Squid	266
The <code>http_port</code> Tag	267
The <code>cache_dir</code> Tag	267
Access Control Lists	269
Configuring SNMP	271
Configuring the <code>cachemgr.cgi</code> Utility	272
New in Squid 2.4—Help for IE Users!	274
Configuring Access to Squid Services	274
The Basics of Basic-Auth	274
Access Control for Users	275
Access Control Lifetime	276
Configuring Proxy Clients	277
Exercise 9.1 Configuring Netscape Navigator	277
Exercise 9.2 Configuring Lynx	278
Exercise 9.3 Configuring Internet Explorer	279
Automatic Proxy Configuration	279
Excluding Access to Restricted Web Sites	281
Filtering Content by URL	281
Filtering by Destination Domain	282
Filtering by MIME Type	282
Filtering by Content-Length Header	283
Summary	284
Solutions Fast Track	284
Frequently Asked Questions	286
Chapter 10 Dissecting Hacks	287
Introduction	288
Securing against Denial of Service Hacks	288
Ping of Death	289
Syn Flood	290
E-Mail Flood	294

Securing against Brute Force Hacks

Like other System VR4 UNIX operating systems, Solaris keeps account information in two files:

- A globally readable `/etc/passwd` file containing noncritical data such as the account name, default shell, user ID, and group ID.
- An `/etc/shadow` file for the account passwords, password expiration dates, and other critical account data.

Securing against Buffer Overflow Hacks	295
Buffer Overflow against a Web Server	302
Buffer Overflow against an FTP Server	305
Securing against Brute Force Hacks	306
Defending against Password Crackers	308
Securing against Trojan Horse Hacks	309
Defending against Rootkits	309
Defusing Logic Bombs	311
Securing cron Jobs	311
Defending against PATH and Command Substitution	313
Securing against IP Spoofing	314
Securing Your <code>.rhosts</code> File	316
MAC Address Spoofing	316
Summary	318
Solutions Fast Track	319
Frequently Asked Questions	321

Chapter 11 Detecting and Denying Hacks 325

Introduction	326
Monitoring for Hacker Activity	326
Using Tripwire	326
The Tripwire Global Settings	328
Tripwire E-Mail Settings	330
Tripwire's Monitored Files	331
Using Shell Scripts to Alert Systems Administrators	335
Monitoring Running Processes	335
Monitoring CPU Activity	337
Putting It All Together	338
What to Do Once You've Detected a Hack	340
What's a Honeytrap?	340
How to Build a Honeytrap on a Sun System	340
Commercial Honeytraps for Solaris	343
Monitoring Solaris Log Files	346
Solaris Log Files to Review	347

Creating Daily Reports

There are many excellent ways to automate the process of reviewing log files. One very popular application is called *swatch*. This application gets its name from the term *simple watcher* and *filter*. It was written in Perl by Todd Adkins and can be found at www.stanford.edu/~atkins/swatch. Swatch is easy to install and configure and can be very helpful in monitoring your log files and alerting you to potential problems.

Didn't You Used to Be Called utmp?	347
The /var/adm/messages File	347
The /var/adm/lastlog File	349
The /etc Files	349
Creating Daily Reports	350
A State-of-the-System Report	350
Headline News	351
The Sports Page	351
Local Events	352
Start the Presses!	353
Summary	357
Solutions Fast Track	358
Frequently Asked Questions	359
Hack Proofing Sun Solaris 8 Fast Track	361
Index	381