

SYNGRESS®



1 YEAR UPGRADE
BUYER PROTECTION PLAN

HACK PROOFING LINUX™

Your Guide to Open Source Security

- Step-by-Step Instructions for Deploying Open Source Security Tools
- Hundreds of Tools & Traps and Damage & Defense Sidebars, Security Alerts, and Exercises!
- Bonus Wallet CD with Configuration Examples, Packet Captures, and Programs

James Stanger, Ph.D.
Patrick T. Lane
Edgar Danielyan Technical Editor



Global Knowledge
RECOMMENDED READING



s o l u t i o n s @ s y n g r e s s . c o m

With more than 1,500,000 copies of our MCSE, MCSD, CompTIA, and Cisco study guides in print, we continue to look for ways we can better serve the information needs of our readers. One way we do that is by listening.

Readers like yourself have been telling us they want an Internet-based service that would extend and enhance the value of our books. Based on reader feedback and our own strategic plan, we have created a Web site that we hope will exceed your expectations.

Solutions@syngress.com is an interactive treasure trove of useful information focusing on our book topics and related technologies. The site offers the following features:

- One-year warranty against content obsolescence due to vendor product upgrades. You can access online updates for any affected chapters.
- “Ask the Author”™ customer query forms that enable you to post questions to our authors and editors.
- Exclusive monthly mailings in which our experts provide answers to reader queries and clear explanations of complex material.
- Regularly updated links to sites specially selected by our editors for readers desiring additional reliable information on key topics.

Best of all, the book you’re now holding is your key to this amazing site. Just go to **www.syngress.com/solutions**, and keep this book handy when you register to verify your purchase.

Thank you for giving us the opportunity to serve your needs. And be sure to let us know if there’s anything else we can do to help you get the maximum value from your investment. We’re listening.

www.syngress.com/solutions



1 YEAR UPGRADE
BUYER PROTECTION PLAN

HACK PROOFING™

Linux: A Guide to Open Source Security

The Only Way to Stop a Hacker Is to Think Like One

James Stanger
Patrick T. Lane

SYNGRESS®

Syngress Publishing, Inc., the author(s), and any person or firm involved in the writing, editing, or production (collectively “Makers”) of this book (“the Work”) do not guarantee or warrant the results to be obtained from the Work.

There is no guarantee of any kind, expressed or implied, regarding the Work or its contents. The Work is sold AS IS and WITHOUT WARRANTY. You may have other legal rights, which vary from state to state.

In no event will Makers be liable to you for damages, including any loss of profits, lost savings, or other incidental or consequential damages arising out from the Work or its contents. Because some states do not allow the exclusion or limitation of liability for consequential or incidental damages, the above limitation may not apply to you.

You should always use reasonable care, including backup and other appropriate precautions, when working with computers, networks, data, and files.

Syngress Media®, Syngress®, and “Career Advancement Through Skill Enhancement®,” are registered trademarks of Syngress Media, Inc. “Ask the Author™,” “Ask the Author UPDATE™,” “Mission Critical™,” and “Hack Proofing™” are trademarks of Syngress Publishing, Inc. Brands and product names mentioned in this book are trademarks or service marks of their respective companies.

KEY	SERIAL NUMBER
001	NFKA4UR934
002	DFTGEGHFG6
003	9456VMPDSP
004	MKC8EWR535
005	ZL94V343BB
006	AS56J89HGE
007	MJTY3D29H6
008	ADQW9UU6NN
009	5TGBXDQ7TN
010	KRF4W2F6P9

PUBLISHED BY
Syngress Publishing, Inc.
800 Hingham Street
Rockland, MA 02370

Hack Proofing Linux: A Guide to Open Source Security

Copyright © 2001 by Syngress Publishing, Inc. All rights reserved. Printed in the United States of America. Except as permitted under the Copyright Act of 1976, no part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written permission of the publisher, with the exception that the program listings may be entered, stored, and executed in a computer system, but they may not be reproduced for publication.

Printed in the United States of America

1 2 3 4 5 6 7 8 9 0

ISBN: 1-928994-34-2

Technical Editors: Edgar Danielyan and Larry Karnis
Co-Publisher: Richard Kristof
Acquisitions Editor: Catherine B. Nolan
Developmental Editor: Kate Glennon
CD Production: Michael Donovan

Freelance Editorial Manager: Maribeth Corona-Evans
Cover Designer: Michael Kavish
Page Layout and Art by: Shannon Tozier
Copy Editor: Beth A. Roberts and Darren Meiss
Indexer: Jennifer Coker

Distributed by Publishers Group West in the United States.



Acknowledgments

We would like to acknowledge the following people for their kindness and support in making this book possible.

Richard Kristof and Duncan Anderson of Global Knowledge, for their generous access to the IT industry's best courses, instructors, and training facilities.

Ralph Troupe, Rhonda St. John, and the team at Callisma for their invaluable insight into the challenges of designing, deploying and supporting world-class enterprise networks.

Karen Cross, Lance Tilford, Meaghan Cunningham, Kim Wylie, Harry Kirchner, Bill Richter, Kevin Votel, and Kent Anderson of Publishers Group West for sharing their incredible marketing experience and expertise.

Mary Ging, Caroline Hird, Simon Beale, Caroline Wheeler, Victoria Fuller, Jonathan Bunkell, and Klaus Beran of Harcourt International for making certain that our vision remains worldwide in scope.

Anneke Baeten, Annabel Dent, and Laurie Giles of Harcourt Australia for all their help.

David Buckland, Wendi Wong, Daniel Loh, Marie Chieng, Lucy Chong, Leslie Lim, Audrey Gan, Charlotte Chan, and Joseph Chan of Transquest Publishers for the enthusiasm with which they receive our books.

Kwon Sung June at Acorn Publishing for his support.

Ethan Atkin at Cranbury International for his help in expanding the Syngress program.

Joe Pisco, Helen Moyer, Paul Zanolli, Alan Steele, and the great folks at InterCity Press for all their help.

Philip Allen at Brewer & Lord LLC for all his work and generosity.



Contributors

Patrick T. Lane (MCSE, MCP+I, MCT, Network+, i-Net+, CIW) is a Content Architect for ProsoftTraining.com, a leading Internet skills training and curriculum development company. He is the author of more than 20 technical courses and is the Director of the CIW Foundations and CIW Internetworking Professional series. While at ProsoftTraining.com, Patrick helped create the Certified Internet Webmaster (CIW) program and the i-Accelerate program for Intel, Novell, and Microsoft professionals.

Patrick consults as a mail, news, FTP, and Web Administrator for several organizations, including jCert Initiative Inc. and ProsoftTraining.com. He is also a network security consultant and writer who specializes in TCP/IP internetworking, LAN/WAN solutions, network and operating system security, and the Linux and Windows NT/2000 platforms. He has consulted for the University of Phoenix/Apollo Group, Novell, Intel, NETg, WAVE technologies, KT Solutions, SmartForce, and Futurekids. Patrick is a member of the CompTIA Network+ Advisory Committee, and co-author of Syngress Publishing's *E-mail Virus Protection Handbook* (ISBN: 1-928994-23-7). His work has been published in eight languages and he has been a featured speaker for the SmartForce Seminar Series on E-Business, the Internet World PING Series on Internet Protocol version 6, and the Information Technology Association of America (ITAA). He holds a master's degree in education.

James Stanger (Ph.D., MCSE, MCT) directs the Linux, Security, and Server Administrator certification tracks for ProsoftTraining.com. Since receiving his Ph.D. in 1997, he has focused on auditing Internet servers and writing courseware, books, and articles about administering and securing Internet servers. James has consulted for IBM, Symantec, Evinci

(www.evinci.org), Pomeroy (www.pomeroy.com), Securify (www.securify.com), Brigham Young University, and California State, San Bernardino. He specializes in troubleshooting firewalls, intrusion detection, DNS, e-mail, and Web server implementations.

James was the Technical Editor of Syngress Publishing's *E-mail Virus Protection Handbook* (ISBN: 1-928994-23-7) and has been an instructional designer of security and A+ courses for NetG, Thompson/WAVE learning, and ComputerPREP. Active in the Linux community, James sits on the Linux Professional Institute (www.lpi.org), SAIR (www.linuxcertification.org), and CompTIA Linux+ (www.comptia.org) advisory boards, each of which is dedicated to creating and maintaining industry-respected certifications. As the Vice Chair of the Linux Professional Institute (LPI) Advisory Council, he acts as liaison between the LPI and companies such as IBM, Compaq, and Intel.



Technical Editors

Edgar Danielyan (CCNA) is a self-employed developer specializing in GCC, X Window, Tcl/Tk, logic programming, Internet security, and TCP/IP; as well as having with BSD, SVR4.2, FreeBSD, SCO, Solaris, and UnixWare. He has a diploma in company law from the British Institute of Legal Executives as well as a paralegal certificate from the University of Southern Colorado. He is currently working as the Network Administrator and Manager of a top-level Armenian domain. He has also worked for the United Nations, the Ministry of Defense of the Republic of Armenia, and Armenian national telephone companies and financial institutions. Edgar speaks four languages, and is a member of ACM, IEEE CS, USENIX, CIPS, ISOC, and IPG.

Larry Karnis (RHCE, Master ACE, CITP), is a Senior Consultant for Application Enhancements, a Unix, Linux, and Internet consulting firm located in Toronto, Canada. His first exposure to Unix was over 20 years ago where he used Unix Version 6 while completing a bachelor's degree in computer science and mathematics. Larry deploys and manages Linux-based solutions such as Web and file and print servers, and Linux firewalls.

About the CD

This book is accompanied by a CD containing files and open source programs used throughout the book. The files include configuration examples, packet captures, and additional resources. We have included the specific open source programs used in the book so you can follow the chapter demonstrations step-by-step on your own systems.

Each file on the CD is discussed in detail and referenced throughout the book with the CD icon below. When a specific file or program is required, it directs you to the accompanying CD. The book also directs you to the Web site where you can download the most current version, and find additional resources relating to that program. For instance, you can download Free Secure Wide Area Network (FreeS/WAN) at www.freeswan.org, or use the version located on the CD. It is recommended that you use the version included on the CD because this will increase the chances that the book demonstrations will be successful, as some of the programs may have changed since this book was printed.

The book is written to Red Hat Linux 7.x. Therefore, most of the CD files are Red Hat Package Manager (.rpm) files. There are also many Tape Archive (.tar) files and GNU Zip (.gzip) files. Instructions for unpacking and installing these files are included in their respective locations throughout the book. To mount the CD onto your Linux system, you would issue the following command (for Red Hat systems):

```
mount -t iso9660 /dev/cdrom /mnt/cdrom
```

And to unmount:

```
umount /mnt/cdrom
```

It is recommended that you copy the CD files to your hard drive before working with them. If you use other versions of Linux, you may need to modify the demonstrations, or download a portable version of the open source programs to work with your version of Linux.



Look for this CD icon when obtaining files used in the book demonstrations.

Contents

Using the GNU General Public License

The GNU General Public License (GPL) is the basis of the open source movement. This license is provided by the Gnu is Not Unix (GNU) organization, which develops various software packages. The most important element of this license is that instead of protecting a particular person or company, it protects the software code that creates the application.

Foreword

xxvii

Chapter 1 Introduction to Open Source Security

1

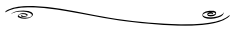
Introduction	2
The Tools Used in This Book	3
Using the GNU General Public License	3
Fee-Based GPL Software	5
Can I Use GPL Software in My Company?	5
Soft Skills: Coping with Open Source Quirks	6
General Lack of Installation and Configuration Support	6
Infrequent or Irregular Update Schedules	6
Command-Line Dominance	6
Lack of Backward Compatibility and No Regular Distribution Body	7
Inconvenient Upgrade Paths	7
Conflicts in Supporting Libraries and Limited Platform Support	7
Interface Changes	8
Partially Developed Solutions	8
Should I Use an RPM or Tarballs?	10
Tarball	10
Red Hat Package Manager	11
Debian	11
Obtaining Open Source Software	12
SourceForge	12
Freshmeat	13
Packetstorm	14

SecurityFocus	15
Is That Download Safe?	16
A Brief Encryption Review	16
Symmetric Key Encryption	17
Asymmetric Key Encryption	18
Public Key and Trust Relationships	19
One-Way Encryption	20
GNU Privacy Guard	21
Deploying GNU Privacy Guard	21
Skipping Public Key Verification	29
Using GPG to Verify Signatures on	
Tarball Packages	30
Using Md5sum	30
Auditing Procedures	31
Locking Down Your Network Hosts	31
Securing Data across the Network	32
Protecting the Network Perimeter	33
Summary	35
Solutions Fast Track	35
Frequently Asked Questions	38

Chapter 2 Hardening the Operating System 41

Introduction	42
Updating the Operating System	42
Red Hat Linux Errata and Update Service	
Packages	42
Handling Maintenance Issues	43
Red Hat Linux Errata: Fixes and Advisories	44
Bug Fix Case Study	46
Manually Disabling Unnecessary Services	
and Ports	47
Services to Disable	47
The xinetd.conf File	48
Locking Down Ports	50
Well-Known and Registered Ports	50
Determining Ports to Block	52

Determining Which Ports to Block



When determining which ports to block on your server, you must first determine which services you require. In most cases, block all ports that are not exclusively required by these services. This is tricky, because you can easily block yourself from services you need, especially services that use ephemeral ports. If your server is an exclusive e-mail server running SMTP and IMAP, you can block all TCP ports except ports 25 and 143, respectively. If your server is an exclusive HTTP server, you can block all ports except TCP port 80.

Blocking Ports	53
Xinetd Services	53
Stand-Alone Services	54
Hardening the System with Bastille	55
Bastille Functions	55
Bastille Versions	63
Implementing Bastille	64
Undoing Bastille Changes	74
Controlling and Auditing Root Access with Sudo	77
System Requirements	79
The Sudo Command	79
Downloading Sudo	80
Installing Sudo	82
Configuring Sudo	86
Running Sudo	90
No Password	92
Sudo Logging	93
Managing Your Log Files	96
Using Logging Enhancers	97
SWATCH	97
Scanlogd	100
Syslogd-ng	101
Summary	103
Solutions Fast Track	104
Frequently Asked Questions	107

Chapter 3 System Scanning and Probing 109

Introduction	110
Scanning for Viruses Using the AntiVir Antiviru	
Application	110
Understanding Linux Viruses	110
Using AntiVir	112
Key Mode and Non-Key Mode	114
Licensing AntiVir	114
Exercise: Updating AntiVir	114
Using TkAntivir	116
Required Libraries and Settings	117

Learn How to Set Preferences For TkAntivir



Scanning Systems for Boot Sector and E-Mail Viruses	117
Additional Information	120
Exercise: Using TkAntivir	120
Scanning Systems for DDoS Attack Software	
Using a Zombie Zapper	123
How Zombies Work and How to Stop Them	124
When Should I Use a Zombie Zapper?	125
What Zombie Zapper Should I Use?	125
What Does Zombie Zapper Require to Compile?	127
Exercise: Using Zombie Zapper	127
Scanning System Ports Using the Gnome Service	
Scan Port Scanner	129
Required Libraries	130
Why Use a Port Scanner?	131
Exercise: Using Gnome Service Scanner	131
Using Nmap	133
Isn't Nmap Just Another Port Scanner?	134
Acquiring and Installing Nmap	136
Common Nmap Options	136
Applied Examples	137
Scanning Entire Networks and Subnets	138
Selective Scanning	139
Adding More Stealth	139
Saving to Text and Reading from Text	140
Testing Firewalls and Intrusion Detection Systems	141
Example: Spoofing the Source Address of a Scan	142
Timing Your Scan Speeds	142
Example: Conducting a Paranoid Scan	143
Exercise: Using Nmap	143
Using Nmap in Interactive Mode	144
Exercise: Using Nmap in Interactive Mode	144

Using NmapFE as a Graphical Front End	146
Exercise: Using NmapFE	147
Using Remote Nmap (Rnmap) as a Central Scanning Device	147
Exercise: Scanning Systems with Rnmap	148
Deploying Cheops to Monitor Your Network	151
How Cheops Works	153
Obtaining Cheops	154
Required Libraries	154
The Cheops Interface	155
Mapping Relations between Computers	157
Cheops Monitoring Methods	157
Connectivity Features	159
Exercise: Installing and Configuring Cheops	160
Deploying Nessus to Test Daemon Security	165
The Nessus Client/Server Relationship	167
Windows Nessus Clients	169
Required Libraries	169
Order of Installation	170
Configuring Plug-Ins	173
Creating a New Nessus User	174
The Rules Database	174
Exercise: Installing Nessus and Conducting a Vulnerability Scan	175
Updating Nessus	179
Understanding Differential, Detached, and Continuous Scans	180
Exercise: Conducting Detached and Differential Scans with Nessus	182
Summary	185
Solutions Fast Track	185
Frequently Asked Questions	189

Chapter 4 Implementing an Intrusion Detection System 191

Introduction	192
Understanding IDS Strategies and Types	194
IDS Types	195
Host-Based IDS Applications	196
Network-Based IDS Applications	196
IDS Applications and Fault Tolerance	197
What Can an IDS Do for Me?	200
Which IDS Strategy Is Best?	203
Network-Based IDS Applications and	
Firewalls	203
IDS Applications	204
Installing Tripwire to Detect File Changes on	
Your Operating System	206
Tripwire Dependencies	207
Availability	208
Deploying Tripwire	208
Tripwire Files	208
Tripwire Installation Steps	209
Configuring the Tripwire Policy File	209
Creating the Tripwire Policy File	212
Database Initialization Mode	212
Testing E-Mail Capability	214
Integrity Checking Mode	214
Specifying a Different Database	215
Reading Reports	215
Updating Tripwire to Account for Legitimate	
Changes in the OS	215
Updating the Policy	216
What Do I Do if I Find a Discrepancy?	217
Configuring Tripwire to Inform You Concerning	
Changes	217
Exercise: Installing Tripwire	217
Exercise: Securing the Tripwire Database	219
Exercise: Using Cron to Run Tripwire	
Automatically	220

SECURITY ALERT!

Although Tripwire has a "file integrity mode," Tripwire is not really an integrity checker in the classic sense. It does not, for example, test the file's stability or inode number or any other aspect in regards to file storage. Tripwire simply compares a file's new signature with that taken when the database was created. Other tools may be used to check the integrity of a file's permissions and ownership information.

Deploying PortSentry to Act as a	
Host-Based IDS	220
Important PortSentry Files	221
Installing PortSentry	222
Configuring PortSentry to Block Users	222
Optimizing PortSentry to Sense Attack Types	223
Exercise: Installing and Configuring	
PortSentry	224
Exercise: Clearing Ipchains Rules	227
Exercise: Running an External Command	
Using PortSentry	227
Installing and Configuring Snort	229
Availability	229
Supporting Libraries	229
Understanding Snort Rules	230
Snort Variables	230
Snort Files and Directories	231
Snort Plug-Ins	232
Starting Snort	233
Logging Snort Entries	236
Running Snort as a Network-Based IDS	236
Ignoring Hosts	237
Additional Logging Options: Text	
files, Tcpdump, and Databases	237
Configuring Snort to Log to a Database	238
Controlling Logging and Alerts	239
Getting Information	240
Exercise: Installing Snort	240
Exercise: Using Snort as an IDS	
Application	241
Exercise: Configuring Snort to Log to	
a Database	243
Exercise: Querying a Snort Database	
from a Remote Host	251
Identifying Snort Add-Ons	251
SnortSnarf	252

Exercise: Using SnortSnarf to Read Snort Logs	252
Analysis Console for Intrusion Databases	252
Summary	254
Solutions Fast Track	254
Frequently Asked Questions	258

Chapter 5 Troubleshooting the Network with Sniffers **261**

Learn the Flags Used in TCP Connections



Flag	Description
SYN	Synchronize sequence numbers. Used for connection establishment.
FIN	The sender is finished with the connection. Used for connection termination.
RST	Reset the connection.
PSH	Push the data.
ACK	Acknowledgment
URG	Urgent

Introduction	262
Understanding Packet Analysis and TCP Handshakes	264
TCP Handshakes	265
Establishing a TCP Connection	265
Terminating a TCP Connection	266
Creating Filters Using Tcpdump	268
Tcpdump Options	268
Tcpdump Expressions	271
Boolean Operators	275
Installing and Using Tcpdump	276
Configuring Ethereal to Capture Network Packets	279
Ethereal Options	281
Ethereal Filters	283
Configuring Ethereal and Capturing Packets	283
Viewing Network Traffic between Hosts Using EtherApe	288
Configuring EtherApe and Viewing Network Traffic	289
Summary	293
Solutions Fast Track	294
Frequently Asked Questions	296

Chapter 6 Network Authentication and Encryption **299**

Introduction	300
Understanding Network Authentication	300

Answer Your Questions about Kerberos

Q: I wish to remove a principal from the keytab of one of my Kerberos clients. How do I do this?

A: Enter `kadmin` as an administrative user on the Kerberos client (not the KDC) and use the `ktremove` option. For example, if you wanted to remove the principal for the user named *james*, you would do the following:

```
terminal$ /usr/
kerberos/sbin/kadmin
kadmin: ktremove
-p james
kadmin: quit
terminal$
```

Attacking Encrypted Protocols	301
Creating Authentication and Encryption Solutions	303
Implementing One-Time Passwords (OTP and OPIE)	305
What Files Does OPIE Replace?	305
How Does OPIE Work?	305
OPIE Files and Applications	306
<code>opiepasswd</code>	307
Password Format	308
Using <code>opiekey</code>	309
Using <code>opieinfo</code> and <code>opiekey</code> to Generate a List	310
Installing OPIE	310
Configuration Options	310
Installation Options	311
Uninstalling OPIE	312
Exercise: Installing OPIE	312
Exercise: Installing the OPIE Client on a Remote Server	315
Exercise: Using <code>opie-tk</code> and Allowing Windows Users to Deploy OPIE.	316
Exercise: Installing <code>opieftpd</code>	318
Implementing Kerberos Version 5	319
Why Is Kerberos Such a Big Deal?	320
Kerberos Terms	321
Kerberos Principals	322
The Kerberos Authentication Process	323
How Information Traverses the Network	324
Creating the Kerberos Database	325
Using <code>kadmin.local</code>	325
Using <code>kadmin</code>	326
Using <code>kadmin</code> on the Client	328
Using <code>kadmin</code> and Creating Kerberos Client Passwords	329
Setting Policies	330
Using <code>Kinit</code>	330

The kinit Command and Time Limits	332
Managing Kerberos Client Credentials	333
The kdestroy Command	333
Exercise: Configuring a KDC	334
Establishing Kerberos Client Trust Relationships with kadmin	337
Additional Daemon Principal Names	339
Logging On to a Kerberos Host Daemon	340
Common Kerberos Client Troubleshooting Issues and Solutions	340
Kerberos Client Applications	341
Kerberos Authentication and klogin	342
Exercise: Configuring a Kerberos Client	342
Summary	345
Solutions Fast Track	345
Frequently Asked Questions	348

Chapter 7 Avoiding Sniffing Attacks through Encryption 353

Secure E-Commerce Transactions

If hackers were alerted to an unsecure server, they could capture packets going in and out of the server to gain the data they sought. For example, if an e-commerce server does not use any type of network encryption for transactions, there is a great deal of data to be gained by a hacker. Unfortunately, many small companies or entrepreneurs set up their own Web servers, unaware of potential security problems, and set up simple scripts to process payment forms.

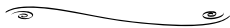
Introduction	354
Understanding Network Encryption	354
Capturing and Analyzing Unencrypted Network Traffic	355
Using OpenSSH to Encrypt Network Traffic between Two Hosts	361
The OpenSSH Suite	362
Installing OpenSSH	364
Configuring SSH	367
How SSH Works	368
Insecure r-command Authentication	368
Secure SSH Authentication	371
Implementing SSH to Secure Data Transmissions over an Insecure Network	373
Distributing the Public Key	376
Capturing and Analyzing Encrypted Network Traffic	381
Summary	385

Solutions Fast Track	386
Frequently Asked Questions	388

Chapter 8 Creating Virtual Private Networks 391

Introduction	392
Secure Tunneling with VPNs	392
Telecommuter VPN Solution	392
Router-to-Router VPN Solution	394
Host-to-Host VPN Solution	395
Tunneling Protocols	395
Explaining the IP Security Architecture	396
Using IPSec with a VPN Tunneling Protocol	400
Internet Key Exchange Protocol	401
Creating a VPN by Using FreeS/WAN	402
Downloading and Unpacking FreeS/WAN	404
Compiling the Kernel to Run FreeS/WAN	407
Recompiling FreeS/WAN into the New Kernel	417
Configuring FreeS/WAN	420
Testing IP Networking	420
Configuring Public Key Encryption for Secure Authentication of VPN Endpoints	424
Starting the Tunnel	434
Capturing VPN Tunnel Traffic	436
Closing the VPN Tunnel	438
Summary	439
Solutions Fast Track	440
Frequently Asked Questions	441

Secure Tunneling with Virtual Private Networks (VPNs)



VPNs provide a private data network over public telecommunication infrastructures, such as the Internet, by providing authentication and encryption through a data “tunnel” between devices. All data transmitted between the devices through the tunnel is secure, regardless of what programs the devices are running.

Chapter 9 Implementing a Firewall with Ipchains and Iptables 445

Introduction	446
Understanding the Need for a Firewall	447
Building a Personal Firewall	449
Understanding Packet Filtering Terminology	450

Understand Essential Linux Firewall Functions

- IP address conservation and traffic forwarding
- Network differentiation
- Protection against denial-of-service, scanning, and sniffing attacks
- IP and port
- Content filtering
- Packet redirection
- Enhanced authentication and encryption
- Supplemented logging

Choosing a Linux Firewall Machine	452
Protecting the Firewall	452
Deploying IP Forwarding and Masquerading	453
Masquerading	456
Configuring Your Firewall to Filter Network Packets	458
Configuring the Kernel	460
Packet Accounting	460
Understanding Tables and Chains in a Linux Firewall	461
Built-In Targets and User-Defined Chains	462
Specifying Interfaces	463
Setting Policies	464
Using Ipchains to Masquerade Connections	467
Iptables Masquerading Modules	468
Using Iptables to Masquerade Connections	468
Iptables Modules	470
Exercise: Masquerading Connections	
Using Ipchains or Iptables	471
Logging Packets at the Firewall	471
Setting Log Limits	472
Adding and Removing Packet Filtering Rules	472
ICMP Types	473
Exercise: Creating a Personal Firewall and Creating a User-Defined Chain	475
Redirecting Ports in Ipchains and Iptables	477
Configuring a Firewall	478
Setting a Proper Foundation	478
Creating Anti-Spoofing Rules	479
Counting Bandwidth Usage	483
Listing and Resetting Counters	484
Setting Type of Service (ToS) in a Linux Router	484
Setting ToS Values in Ipchains and Iptables	486
Using and Obtaining Automated Firewall Scripts and Graphical Firewall Utilities	488

Firewall Works in Progress	490
Exercise: Using Firestarter to Create a Personal Firewall	490
Exercise: Using Advanced Firestarter Features	498
Summary	500
Solutions Fast Track	500
Frequently Asked Questions	505

Chapter 10 Deploying the Squid Web Proxy Cache Server 507

Introduction	508
Benefits of Proxy Server Implementation	508
Proxy Caching	508
Network Address Translation	510
Differentiating between a Packet Filter and a Proxy Server	512
Implementing the Squid Web Proxy Cache Server	513
System Requirements Specific to Proxy Caching	516
Installing Squid	517
Configuring Squid	520
The http_port Tag	522
The Cache_dir Tag	523
The acl Tag	525
The http_access Tag	526
Starting and Testing Squid	528
Configuring Proxy Clients	529
Configuring Netscape Navigator and Lynx	530
Configuring Netscape Navigator	530
Configuring Lynx	532
Configuring Internet Explorer (Optional)	533
Summary	535
Solutions Fast Track	536
Frequently Asked Questions	538

**Configure Squid
with the /etc/squid/
squid.conf file**



Chapter 11 Maintaining Firewalls 543

Introduction	544
Testing Firewalls	544
IP Spoofing	546
Open Ports/Daemons	546
Monitoring System Hard Drives, RAM, and Processors	547
Suspicious Users, Logins, and Login Times	547
Check the Rules Database	548
Verify Connectivity with Company Management and End Users	548
Remain Informed Concerning the Operating System	549
Port Scans	549
Using Telnet, Ipchains, Netcat, and SendIP to Probe Your Firewall	550
Ipchains	551
Telnet	551
Using Multiple Terminals	552
Netcat	552
Sample Netcat Commands	554
Additional Netcat Commands	555
Exercise: Using Netcat	557
SendIP: The Packet Forger	558
SendIP Syntax	558
Exercise: Using SendIP to Probe a Firewall	560
Understanding Firewall Logging, Blocking, and Alert Options	563
Firewall Log Daemon	563
Obtaining Firelogd	563
Syntax and Configuration Options	563
Message Format	564
Customizing Messages	566
Reading Log Files Generated by Other Firewalls	568

See How to Use the Firelogd Program



Firelogd (Firewall Log Daemon) is a relatively simple program that can either be run as an application or (you might have guessed) as a daemon. It does two things:

- It reads the kernel log entries and passes them into a "first in, first out" (FIFO) pipe, which *Firelogd* can then process.
- Once its buffer is full, it e-mails a report of suspicious traffic to an account of your choosing. You can have it mailed to a local account, or to a remote system of your choice.

Exercise: Configuring and Compiling Firelogd	568
Fwlogwatch	569
Fwlogwatch Modes	570
Fwlogwatch Options and Generating Reports	572
Exercise: Generating an HTML-Based Firewall Log with Fwlogwatch	575
Automating Fwlogwatch	575
The Fwlogwatch Configuration File	576
Notification Options	579
Response Options	581
Exercise: Configuring Fwlogwatch to Send Automatic Alerts and Block Users	583
Using Fwlogwatch with CGI Scripts	584
Obtaining More Information	586
Viewing the Results	587
Exercise: Using Cron and Fwlogwatch CGI Scripts to Generate an Automatic HTML Report	588
Additional Fwlog Features	590
Obtaining Additional Firewall Logging Tools	590
Summary	593
Solutions Fast Track	593
Frequently Asked Questions	597
Appendix A Bastille Log	599
Appendix B Hack Proofing Linux Fast Track	605
Index	637

Preface

Hack Proofing Linux: A Guide to Open Source Security is designed to help you deploy a Linux system on the Internet in a variety of security roles. This book provides practical instructions and pointers concerning the open source security tools that we use every day.

First, we show you how to obtain the software; and then, how to use the Bastille application to “harden” your Linux operating system so that it can function securely as it fulfills a specific role of your choice (e.g., as a Web server, as an E-mail server, and so forth). You will also learn how to use your Linux system as an auditing tool to scan systems for vulnerabilities as well as create an Intrusion Detection System (IDS), which enables your Linux system to log and respond to suspicious activity. From virus protection to encrypting transmissions using Gnu Privacy Guard and FreeSWAN, you will be able to configure your system to secure local data as well as data that will be passed along the network. After reading this book, you will be able to identify open source and “for-fee” tools that can help you further secure your Linux system.

We have also included chapters concerning ways to sniff and troubleshoot network connections and how to implement strong authentication using One Time Passwords (OTP) and Kerberos. Tools such as Squid proxy server and Ipchains/Iptables will help you use your Linux system so that it can act as a firewall. With the tools on the accompanying CD as well as the advice and instructions given in this book, you will be able to deploy your Linux system in various roles with confidence.

We decided to focus on profiling the most commonly used security tools found on the Linux platform. We also decided to emphasize the real-world implementation of these tools, as opposed to just providing conceptual overviews. Finally, we decided to describe the steps you should take when things go wrong. As a result, we have created a book that is a valuable resource that helps you use your Linux system as efficiently as possible.

One of the most exciting things about this book is that it provides hands-on instructions for implementing security applications. From Gnu Privacy Guard (GPG) and Bastille to FreeSWAN, Kerberos, and firewall troubleshooting utilities, this book shows you how to use your Linux skills to provide the most important security services such as encryption, authentication, access control, and logging.

While writing the book, we had the following three-part structure in mind:

- Locking Down the Network (Chapters 1 through 4)
- Securing Data Passing Across the Network (Chapters 5 through 8)
- Protecting the Network Perimeter with Firewalls (Chapters 9 through 11)

Each of these sections is designed to help you find the best solution for your particular situation. Although the book itself isn't explicitly divided into sections, as you are reading remember this rough division because it will help you to implement security measures in your own environment.

Chapter 1 discusses open source concepts, including the GNU General Public License, as presented by the www.gnu.org people (the Free Software Foundation), and then moves on to showing how you can use GPG and Pretty Good Privacy (PGP) to encrypt transmissions and also to check the signatures of files that you download from the Web. It also provides information concerning the steps to take when auditing a network.

Chapter 2 shows you how to lock down your operating system so that it provides only those Internet services that you desire. Chapter 3 shows you how to use applications such as AntiVir, Gnome ServiceScan, Nmap, Rnmap, and Nessus to scan for vulnerabilities. In Chapter 4, you will learn about host and network-based IDS applications such as Snort, Tripwire, and PortSentry. Chapter 5 explains how to use network sniffers such as Tcpdump, Ethereal, and EtherApe to their full advantage. With this knowledge, auditing a network and truly understanding what is going on "beneath the hood" will make you a much more effective network security administrator.

By the time you finish Chapter 6, you will know how to deploy One Time Passwords and Kerberos, and in Chapter 7, you will understand how to avoid sniffing attacks, and in Chapter 8, you will enable IPsec by deploying FreeSWAN. Chapter 9 empowers you to create personal firewalls as well as packet filtering firewalls using either Ipchains or Iptables. Chapter 10 shows you how to implement Squid so that you can more carefully monitor and process packets. Finally, Chapter 11 provides you with tools that test your firewall implementation.

The open source community has fulfilled the need for a powerful, free system that allows you to conduct audits, serve up Web pages, provide e-mail services, or any other Internet service you wish to provide. Once you are able to take advantage of the security software provided by the open source community, you will receive the benefit of having a huge pool of developers working for you. You will gain more freedom because you will be able to choose widely tested security tools provided by a variety of skilled developers. You can even choose (at your own risk) to use rather obscure tools that have been recently created. It is up to you.

Open source operating systems and security tools are both a blessing and a curse: You are blessed with (usually) free software, but you are then cursed with having to spend time working with the software's idiosyncrasies. By reading this book and implementing the tools and practices we've described, you should be able to minimize the "curse." It is also our hope that as you read this book you will also become further involved in the open source software movement, which has begun to fulfill its promise of creating powerful, useful software.

—James Stanger, Ph.D., MCSE, MCT

Introduction to Open Source Security

Solutions in this chapter:

- Using the GNU General Public License
- Soft Skills: Coping with Open Source Quirks
- Should I Use an RPM or Tarballs?
- Obtaining Open Source Software
- A Brief Encryption Review
- Public Key and Trust Relationships
- Auditing Procedures

- ☑ Summary
- ☑ Solutions Fast Track
- ☑ Frequently Asked Questions

Introduction

In spite of the ups and downs of the dot-com industry, open source software has become a viable alternative to commercial companies such as Microsoft, Sun, and IBM. Although open source software has its quirks and its problems, the open source movement has made its niche in the networking market. As a networking professional, it is in your best interest to understand some of the more important security applications and services that are available.

This book is designed to provide experienced systems administrators with open source security tools. Although we have made every effort to include as many people and as many skill sets as possible, this book assumes a fundamental knowledge of Linux. This book focuses on open source Linux applications, daemons, and system fixes. In the book's first chapters, you will learn how to lock down your network. Chapter 2 discusses ways to secure and monitor the operating system, and ways to scan local and remote networks for weaknesses. You will receive detailed information on how to ensure that your system's services and the root account are as secure as possible.

In Chapter 3, you will learn how to deploy antivirus and scanning programs for your local system. By using these scanning programs, you will be able to mitigate risk and learn more about the nature of services on your network. Scanners such as nmap and nessus will help you learn about the open ports on your network, and how these open ports might pose a threat to your system. Chapter 3 gives you detailed information about practical ways to implement intrusion detection on your local system and on your network. Using applications such as Tripwire, Portsentry, and Snort, you will be able to precisely identify system anomalies and detect inappropriate logins. Chapter 5 shows how you can use open source tools such as tcpdump, Ethereal, EtherApe, and Ntop to inspect and gauge traffic on the network.

The second part of the book focuses on ways to enhance authentication using open source software. In Chapter 6, you will learn about One Time Passwords (OTP) and Kerberos as ways to ensure that malicious users won't be able to obtain your passwords as they cross the network. Chapter 7 discusses ways to use Secure Shell (SSH) and Secure Sockets Layer (SSL), which are ways to enable on-the-fly encryption to protect data. In Chapter 8, you will learn about how to enable IPsec on a Linux system so that you can implement a virtual private network (VPN). As you learn more about the primary VPN product called Free Secure Wide Area Network (FreeS/WAN), you will see how it is possible to protect network traffic as it passes through your own network, and over the Internet.

The final part of the book focuses on ways to create an effective network perimeter. Chapter 9, shows how to install and configure Ipchains and Iptables on a Linux system. Kernels earlier than 2.3 can use Ipchains, whereas kernel versions 2.3 and later use Iptables. Regardless of the way you do it, you will learn to filter traffic with these two packet filtering tools.

In Chapter 10, you will learn how a proxy server can further enhance your control over your network perimeter. Specifically, you will use the Squid proxy server to control client access to the Internet. You will also learn how to configure Linux clients to access the proxy server. Finally, Chapter 11, shows how to troubleshoot and counteract problems with your network perimeter. You will learn how to maintain, test, and log the firewall so that you have a functional barrier between you and the outside world.

It is our intention to create a book that gives you practical information and advice about the most common open source security tools.

The Tools Used in This Book

This book was written using version 7.0 of the Red Hat Linux operating system. Although it may not be the “best” Linux distribution (there are at least 100 versions in the world), it is the most popular. We have tried to ensure that the skills and tools you obtain in this book will be portable to other Linux versions, and even other open source operating systems such as FreeBSD (www.freebsd.org). However, each Linux flavor has its own quirks, and you may find it necessary to deviate from some of the instructions in this book.

Using the GNU General Public License

The GNU General Public License (GPL) is the basis of the open source movement. This license is provided by the Gnu is Not Unix (GNU) organization, which develops various software packages. Begun in 1984 by Richard Stallman, GNU has worked to create a license designed to ensure that the open source movement continues to thrive. You can learn more about GNU at the www.gnu.org Web site, shown in Figure 1.1.

The most important element of this license is that instead of protecting a particular person or company, it protects the software code that creates the application. Traditionally, copyrights have enabled individuals to lay claim to a particular piece of software and then sell it for profit. In addition, the copyright enables that individual to then take action against anyone else who uses that code to create

similar functionality. For better or for worse, Richard Stallman, Eric Raymond, and others helped found and popularize the concept of an open software license called the Gnu General Public License (often referred to as the *GPL*). You can read the GPL at www.gnu.org/copyleft/gpl.html.

Figure 1.1 The GNU Web Site



This license is part of the “copyleft” movement, which considers itself an alternative to traditional copyright laws. The GPL essentially allows anyone who develops code to ensure that the code remains open, meaning that GPL-licensed code can be taken and improved upon by anyone, as long as the improved code is given to the original writer and the software writing community. Consequently, a piece of code protected by the GPL will, by law, always remain accessible by anyone who wants to read or modify it. Without the GPL license, another person can take the code that you invent, and make it closed and proprietary.

The GNU GPL is not the only free software license in existence. Figure 1.2 shows the GNU page dedicated to understanding additional licenses. If you wish, you can read about additional licenses that are similar to the GPL at www.gnu.org/philosophy/license-list.html.

For more information about the open source movement, one of the more revealing books is Erik Raymond’s *The Cathedral and the Bazaar* (O’Reilly &

Associates, 2001). Although somewhat overly enthusiastic, it is a very helpful book in understanding the mindset of many open source code writers.

Figure 1.2 Viewing GNU's Licenses Comment Section



Fee-Based GPL Software

Contrary to what you might think, open source code protected by the GPL is not necessarily free. Under the terms of the GPL, any person or corporation can take GPL software, modify it, and then package it for sale. However, this person or corporation must make this software freely available for anyone to read or modify.

Can I Use GPL Software in My Company?

The GNU GPL does not ask companies to supply licensing agreements or otherwise register the programs. However, other licenses, which you can read at GNU's comparative forum, may invoke restrictions that you may have to consider as you implement the software. The software covered in this book is, in one way or another, open software, which means it can be used by any organization.