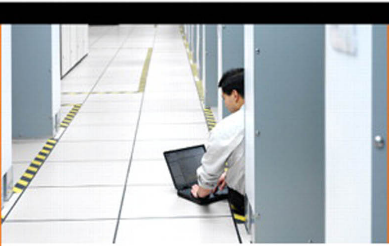


SECURITY

Penetration Testing and Network Defense

The practical guide to simulating, detecting, and responding to network attacks

Andrew Whitaker
Daniel Newman

ciscopress.com

FREE SAMPLE CHAPTER

SHARE WITH OTHERS

Penetration Testing and Network Defense

Andrew Whitaker, Daniel P. Newman

Cisco Press
800 East 96th Street
Indianapolis, IN 46240 USA

Penetration Testing and Network Defense

Andrew Whitaker and Daniel P. Newman

Copyright© 2006 Cisco Systems, Inc.

Published by:
Cisco Press
800 East 96th Street
Indianapolis, IN 46240 USA

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without written permission from the publisher, except for the inclusion of brief quotations in a review.

Printed in the United States of America 5 6 7 8 9 0

Fifth Printing September 2010

Library of Congress Cataloging-in-Publication Number: 2004108262

ISBN: 1-58705-208-3

Warning and Disclaimer

This book is designed to provide information about penetration testing and network defense techniques. Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied.

The information is provided on an “as is” basis. The authors, Cisco Press, and Cisco Systems, Inc. shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or from the use of the discs or programs that may accompany it.

The opinions expressed in this book belong to the authors and are not necessarily those of Cisco Systems, Inc.

Feedback Information

At Cisco Press, our goal is to create in-depth technical books of the highest quality and value. Each book is crafted with care and precision, undergoing rigorous development that involves the unique expertise of members from the professional technical community.

Readers' feedback is a natural continuation of this process. If you have any comments regarding how we could improve the quality of this book or otherwise alter it to better suit your needs, you can contact us through e-mail at feedback@ciscopress.com. Please make sure to include the book title and ISBN in your message.

Trademark Acknowledgments

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Cisco Press or Cisco Systems, Inc. cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

We greatly appreciate your assistance.

Publisher	John Wait
Editor-in-Chief	John Kane
Cisco Representative	Anthony Wolfenden
Cisco Press Program Manager	Jeff Brady
Executive Editor	Brett Bartow
Production Manager	Patrick Kanouse
Senior Development Editor	Christopher Cleveland
Project Editor	Marc Fowler
Copy Editor	Karen A. Gill
Technical Editors	Steve Kalman, Michael Overstreet
Team Coordinator	Tammi Barnett
Book/Cover Designer	Louisa Adair
Compositor	Mark Shirar
Indexer	Tim Wright

Cisco Systems

Corporate Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 526-4100

European Headquarters
Cisco Systems International BV
Haarlerbergpark
Haarlerbergweg 13-19
1101 CH Amsterdam
The Netherlands
www-europe.cisco.com
Tel: 31 0 20 357 1000
Fax: 31 0 20 357 1100

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
www.cisco.com
Tel: 408 526-7660
Fax: 408 527-0883

Asia Pacific Headquarters
Cisco Systems, Inc.
Capital Tower
168 Robinson Road
#22-01 to #29-01
Singapore 068912
www.cisco.com
Tel: +65 6317 7777
Fax: +65 6317 7799

Cisco Systems has more than 200 offices in the following countries and regions. Addresses, phone numbers, and fax numbers are listed on the **Cisco.com Web site at www.cisco.com/go/offices.**

Argentina • Australia • Austria • Belgium • Brazil • Bulgaria • Canada • Chile • China PRC • Colombia • Costa Rica • Croatia • Czech Republic
Denmark • Dubai, UAE • Finland • France • Germany • Greece • Hong Kong SAR • Hungary • India • Indonesia • Ireland • Israel • Italy
Japan • Korea • Luxembourg • Malaysia • Mexico • The Netherlands • New Zealand • Norway • Peru • Philippines • Poland • Portugal
Puerto Rico • Romania • Russia • Saudi Arabia • Scotland • Singapore • Slovakia • Slovenia • South Africa • Spain • Sweden
Switzerland • Taiwan • Thailand • Turkey • Ukraine • United Kingdom • United States • Venezuela • Vietnam • Zimbabwe

Copyright © 2003 Cisco Systems, Inc. All rights reserved. CCIP, CCSP, the Cisco Arrow logo, the Cisco *Powered* Network mark, the Cisco Systems Verified logo, Cisco Unity, Follow Me Browsing, FormShare, iQ Net Readiness Scorecard, Networking Academy, and ScriptShare are trademarks of Cisco Systems, Inc.; Changing the Way We Work, Live, Play, and Learn, The Fastest Way to Increase Your Internet Quotient, and iQuick Study are service marks of Cisco Systems, Inc.; and Aironet, ASIST, BPX, Catalyst, CCDA, CCDP, CCIE, CCNA, CCNP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, the Cisco IOS logo, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Empowering the Internet Generation, Enterprise/Solver, EtherChannel, EtherSwitch, Fast Step, GigaStack, Internet Quotient, IOS, IP/TV, iQ Expertise, the iQ logo, LightStream, MGX, MICA, the Networkers logo, Network Registrar, *Packet,* PIX, Post-Routing, Pre-Routing, RateMUX, Registrar, SlideCast, SMARTnet, StrataView Plus, Stratm, SwitchProbe, TeleRouter, TransPath, and VCO are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and certain other countries.

All other trademarks mentioned in this document or Web site are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0303R)

Printed in the USA

About the Authors

Andrew Whitaker has been working in the IT industry for more than ten years, specializing in Cisco and security technologies. Currently, he works as the Director of Enterprise InfoSec and Networking for TechTrain, an international computer training and consulting company. Andrew performs penetration testing and teaches ethical hacking and Cisco courses throughout the United States and Europe. Prior to teaching, Whitaker was performing penetration tests for financial institutions across the southeastern United States. He also was previously employed as a senior network engineer with an online banking company, where he was responsible for network security implementation and data communications for e-finance websites. He is certified in the following: CCSP, CCNP, CCNA, CCDA, InfoSec, MCSE, CNE, A+, CNE, Network+, Security+, CEH, and CEI.

Daniel P. Newman has been in the computer industry for more than twelve years specializing in application programming, database design, and network security for projects all over the world. Daniel has implemented secure computer and network solutions to a wide variety of industries ranging from titanium plants, diamond mines, and robotic-control systems to secure Internet banking. Working across four continents, he has gained expertise providing secure computer network solutions within a wide range of systems. Daniel is currently working as a freelance penetration tester and a senior technical trainer teaching Cisco and Microsoft products. In addition, Newman specializes in practicing and training certified ethical hacking and penetration testing. In his pursuit of increased knowledge, he has become certified in the following: A+, Network+, I-Net+, Server+, Linux+, Security+, MCDST, MCSA, MCSE (NT, 2000, 2003); Security, MCDBA, MCT, CCNA, CCDA, CSS1, CCSP, InfoSec, CEH, CEI, and CISSP. In his off time, Newman has authored books on PIX Firewall and Cisco IDS and worked as technical editor for books on the Cisco SAFE model.

About the Technical Reviewers

Stephen Kalman is a data security trainer. He is the author or tech editor of more than 20 books, courses, and CBT titles. His most recent book is *Web Security Field Guide*, published by Cisco Press. In addition to those responsibilities, he runs a consulting company, Esquire Micro Consultants, that specializes in network security assessments and forensics.

Kalman holds CISSP, CEH, CHFI, CCNA, CCDA, A+, Network+, and Security+ certifications and is a member of the New York State Bar.

Michael Overstreet is a delivery manager for Cisco Advanced Services within World Wide Security Practice. He is responsible for the delivery of security assessment and implementation services with a focus on Security Posture Assessments (SPA). He has worked for Cisco for six years delivering the security services. He is a graduate of Christopher Newport University with a Bachelor of Science in Computer Science. Michael holds CISSP and CCNP certifications.

Dedications

Andrew Whitaker:

I dedicate this book in memory of Dr. Bill R. Owens and Dr. Charles Braak. Your legacies continue to inspire me to pursue higher levels of excellence.

And to my amazing wife, Jennifer.

-BFF-

Daniel Newman:

I dedicate this book to my beautiful wife, Clare. No matter how close you are, there is never a moment that you are not in my thoughts and never a time that my heart is not missing you. You are the light of my life that never stops shining brighter and brighter as time goes on. I just wish forever were not so short, because I'll miss you when it comes.

—Your husband, Daniel

Acknowledgments

Andrew Whitaker:

Many people were involved in the creation of this book. First, I must thank my forever supportive wife, whose encouragement kept me focused and motivated to complete this project. You haven't seen much of me this past year, and I thank you for your sacrifice so that I could pursue this book. I will always love you.

To Dan Newman, my coauthor: I can only say thank you for being a great friend and colleague. Despite the long distance between us, you still remain a good friend, and I look forward to working with you on future projects. The dawn is coming!

Two people who deserve special mention are Brett Bartow and Chris Cleveland. You both have saint-like patience to allow for our habitual tardiness.

Acknowledgements must also be given to our two technical editors, Steve Kalman and Michael Overstreet. Steve, without you, this book never would have happened. We are lucky to have you as an editor. Michael, thank you for holding such a high standard to ensure that this book is of quality material.

Several others must be mentioned for their assistance with certain chapters. Jonathan Irvin and Robert Hall at Defcon-5 both shared their social engineering tactics for Chapter 4. For our chapter on buffer overflows, I am very grateful for SolarIce at #CovertSystems, who chatted online with me at 4:00 a.m. one Saturday morning to discuss his exploit techniques. Susan Brenner at the University of Dayton helped with the discussion on cybercrime and ethics in Chapter 2. Susan, your students are lucky to have you.

Still others had an indirect involvement with this book. I'd like to thank John Almeter at NetTek, a man of great integrity who got me started in this field. I also must thank Rick Van Luvender at InfoSec Academy for teaching me so much about penetration testing. Thanks also to the Indian River Starbucks for providing me with a second office.

Finally, I must thank God, for without you, there would be no ethics or morality.

Daniel Newman:

I would like to thank Brett Bartow and Christopher Cleveland for their encouragement, drive, and push to help us keep this massive project on schedule and on time. Thanks, guys!

To our technical editors, Michael Overstreet and Steve Kalman, for double-checking all our facts and helping us fix all our minor typos.

To Andrew, with whom I coauthored this book. Thank you for your never-ending patience with busy work schedules, time zones, and deadlines that plagued us. If only there were 25 hours in the day, we could accomplish so much more. You are the best of friends, and I would like to thank you for the opportunity to work with you on this project—I can't wait to do 167.

I would also like to thank Hannah "Wee" for putting up with Mom and I while we string the den with cables and hammer away on computer keyboards attacking systems for hours on end. You always seem to find a way to still be involved, whether it be getting coffee or just staying close by watching movies on the laptop. Thanks, Wee!

Lastly and most importantly, I would like to thank my wife, Clare. Thank you, honey, for your never-ending patience, technical editing, case study testing, reference checking, and moral support on this book. You are my best friend, my peer, my partner, and my soul mate for life. For without you, this book never would have been possible. I love you, my wonderful partner.

Contents at a Glance

Foreword xxii

Introduction xxiii

Part I Overview of Penetration Testing 3

Chapter 1 Understanding Penetration Testing 5

Chapter 2 Legal and Ethical Considerations 21

Chapter 3 Creating a Test Plan 35

Part II Performing the Test 47

Chapter 4 Performing Social Engineering 49

Chapter 5 Performing Host Reconnaissance 77

Chapter 6 Understanding and Attempting Session Hijacking 127

Chapter 7 Performing Web Server Attacks 177

Chapter 8 Performing Database Attacks 247

Chapter 9 Password Cracking 279

Chapter 10 Attacking the Network 321

Chapter 11 Scanning and Penetrating Wireless Networks 349

Chapter 12 Using Trojans and Backdoor Applications 367

Chapter 13 Penetrating UNIX, Microsoft, and Novell Servers 439

Chapter 14 Understanding and Attempting Buffer Overflows 461

Chapter 15 Denial-of-Service Attacks 481

Chapter 16 Case Study: A Methodical Step-By-Step Penetration Test 501

Part III Appendixes 533

Appendix A Preparing a Security Policy 535

Appendix B Tools 547

Glossary 571

Index 583

Contents

Foreword xxii

Introduction xxiii

Part I Overview of Penetration Testing 3

Chapter 1 Understanding Penetration Testing 5

Defining Penetration Testing 5

Assessing the Need for Penetration Testing 8

Proliferation of Viruses and Worms 9

Wireless LANs 9

Complexity of Networks Today 10

Frequency of Software Updates 10

Availability of Hacking Tools 10

The Nature of Open Source 11

Reliance on the Internet 11

Unmonitored Mobile Users and Telecommuters 12

Marketing Demands 12

Industry Regulations 12

Administrator Trust 13

Business Partnerships 13

Hacktivism 13

Attack Stages 13

Choosing a Penetration Testing Vendor 14

Preparing for the Test 16

Summary 17

Chapter 2 Legal and Ethical Considerations 21

Ethics of Penetration Testing 21

Laws 23

U.S. Laws Pertaining to Hacking 24

1973 U.S. Code of Fair Information Practices 25

1986 Computer Fraud and Abuse Act (CFAA) 25

State Laws 27

Regulatory Laws 28

1996 U.S. Kennedy-Kasselbaum Health Insurance Portability and Accountability Act (HIPAA) 28

Graham-Leach-Bliley (GLB) 29

USA PATRIOT ACT 30

2002 Federal Information Security Management Act (FISMA) 30

2003 Sarbanes-Oxley Act (SOX) 30

Non-U.S. Laws Pertaining to Hacking 31

Logging 31

To Fix or Not to Fix 32

Summary 32

Chapter 3 Creating a Test Plan 35

Step-by-Step Plan 35

Defining the Scope 36

Social Engineering 36

Session Hijacking 36

Trojan/Backdoor 37

Open-Source Security Testing Methodology Manual 37

Documentation 40

Executive Summary 40

Project Scope 42

Results Analysis 42

Summary 44

Appendixes 44

Summary 44

Part II Performing the Test 47

Chapter 4 Performing Social Engineering 49

Human Psychology 50

Conformity Persuasion 50

Logic Persuasion 51

Need-Based Persuasion 52

Authority-Based Persuasion 53

Reciprocation-Based Social Engineering 53

Similarity-Based Social Engineering 54

Information-Based Social Engineering 54

What It Takes to Be a Social Engineer 55

Using Patience for Social Engineering 56

Using Confidence for Social Engineering 57

Using Trust for Social Engineering 58

Using Inside Knowledge for Social Engineering 59

First Impressions and the Social Engineer 60

Tech Support Impersonation 61

Third-Party Impersonation 62

E-Mail Impersonation 64

End User Impersonation 69

Customer Impersonation 69

Reverse Social Engineering 70
Protecting Against Social Engineering 71
Case Study 72
Summary 75

Chapter 5 Performing Host Reconnaissance 77
Passive Host Reconnaissance 78
A Company Website 79
EDGAR Filings 87
NNTP USENET Newsgroups 87
User Group Meetings 88
Business Partners 88
Active Host Reconnaissance 89
NSLookup/Whois Lookups 89
SamSpade 92
Visual Route 95
Port Scanning 96
TCP Connect() Scan 98
SYN Scan 99
NULL Scan 99
FIN Scan 100
ACK Scan 100
Xmas-Tree Scan 101
Dumb Scan 101
NMap 102
NMap Switches and Techniques 103
Compiling and Testing NMap 105
Fingerprinting 106
Footprinting 107
Detecting a Scan 109
Intrusion Detection 109
Anomaly Detection Systems 109
Misuse Detection System 109
Host-Based IDSs 110
Network-Based IDSs 110
Network Switches 111
Examples of Scan Detection 112
Detecting a TCP Connect() Scan 113
Detecting a SYN Scan 114
Detecting FIN, NULL, and Xmas-Tree Scans 115
Detecting OS Guessing 117

Case Study 118
Summary 122
Chapter 6 Understanding and Attempting Session Hijacking 127
Defining Session Hijacking 127
Nonblind Spoofing 128
Blind Spoofing 129
TCP Sequence Prediction (Blind Hijacking) 130
Tools 131
Juggernaut 131
Hunt 134
TTY-Watcher 136
T-Sight 136
Other Tools 137
Beware of ACK Storms 137
Kevin Mitnick's Session Hijack Attack 139
Detecting Session Hijacking 143
Detecting Session Hijacking with a Packet Sniffer 145
Configuring Ethereal 145
Watching a Hijacking with Ethereal 147
Detecting Session Hijacking with Cisco IDS 153
Signature 1300: TCP Segment Overwrite 156
Signature 3250: TCP Hijack 157
Signature 3251: TCP Hijacking Simplex Mode 162
Watching a Hijacking with IEV 164
Protecting Against Session Hijacking 167
Case Study 168
Summary 173
Resources 174
Chapter 7 Performing Web Server Attacks 177
Understanding Web Languages 177
HTML 179
DHTML 181
XML 183
XHTML 184
JavaScript 185
JScript 186
VBScript 186
Perl 187

ASP 188
CGI 191
PHP Hypertext Preprocessor 192
ColdFusion 193
Java Once Called Oak 193
Client-Based Java 194
Server-Based Java 194
Website Architecture 196
E-Commerce Architecture 198
Apache HTTP Server Vulnerabilities 199
IIS Web Server 199
Showcode.asp 200
Privilege Escalation 201
Buffer Overflows 202
Web Page Spoofing 203
Cookie Guessing 205
Hidden Fields 207
Brute Force Attacks 209
Brutus 211
HTTP Brute Forcer 211
Detecting a Brute Force Attack 212
Protecting Against Brute Force Attacks 215
Tools 217
NetCat 217
Vulnerability Scanners 218
IIS Xploit 221
execiis-win32.exe 221
CleanIISLog 222
IntelliTamper 222
Web Server Banner Grabbing 223
Hacking with Google 224
Detecting Web Attacks 225
Detecting Directory Traversal 226
Detecting Whisker 228
Protecting Against Web Attacks 232
Securing the Operating System 232
Securing Web Server Applications 234
IIS 234
Apache 236

Securing Website Design 236
Securing Network Architecture 237
Case Study 238
Summary 244
Chapter 8 Performing Database Attacks 247
Defining Databases 249
Oracle 250
Structure 250
SQL 250
MySQL 251
Structure 251
SQL 251
SQL Server 252
Structure 252
SQL 253
Database Default Accounts 253
Testing Database Vulnerabilities 253
SQL Injection 256
System Stored Procedures 257
xp_cmdshell 259
Connection Strings 259
Password Cracking/Brute Force Attacks 260
Securing Your SQL Server 261
Authentication 261
Service Accounts 263
Public Role 263
Guest Account 264
Sample Databases 264
Network Libraries 264
Ports 265
Detecting Database Attacks 266
Auditing 266
Failed Logins 268
System Stored Procedures 269
SQL Injection 270
Protecting Against Database Attacks 270
Case Study 272
Summary 277
References and Further Reading 277

Chapter 9 Password Cracking 279

Password Hashing 280

Using Salts 282

Microsoft Password Hashing 282

UNIX Password Hashing 284

Password-Cracking Tools 284

John the Ripper 285

Pwdump3 287

L0phtcrack 289

Nutcracker 298

Hypnopædia 299

Snadboy Revelation 300

Boson GetPass 302

RainbowCrack 303

Detecting Password Cracking 305

Network Traffic 306

System Log Files 306

Account Lockouts 307

Physical Access 308

Dumpster Diving and Key Logging 308

Social Engineering 308

Protecting Against Password Cracking 309

Password Auditing 309

Logging Account Logins 309

Account Locking 311

Password Settings 311

Password Length 312

Password Expiration 312

Password History 313

Physical Protection 313

Employee Education and Policy 315

Case Study 316

Summary 319

Chapter 10 Attacking the Network 321

Bypassing Firewalls 321

Evading Intruder Detection Systems 323

Testing Routers for Vulnerabilities 324

CDP 324

HTTP Service 326

Password Cracking 328
Modifying Routing Tables 329
Testing Switches for Vulnerabilities 333
VLAN Hopping 333
Spanning Tree Attacks 334
MAC Table Flooding 335
ARP Attacks 335
VTP Attacks 336
Securing the Network 337
Securing Firewalls 337
Securing Routers 338
Disabling CDP 338
Disabling or Restricting the HTTP Service 338
Securing Router Passwords 338
Enabling Authentication for Routing Protocols 339
Securing Switches 341
Securing Against VLAN Hopping 341
Securing Against Spanning Tree Attacks 341
Securing Against MAC Table Flooding and ARP Attacks 341
Securing Against VTP Attacks 342
Case Study 342
Summary 347
Chapter 11 Scanning and Penetrating Wireless Networks 349
History of Wireless Networks 349
Antennas and Access Points 350
Wireless Security Technologies 351
Service Set Identifiers (SSIDs) 351
Wired Equivalent Privacy (WEP) 352
MAC Filtering 352
802.1x Port Security 352
IPSec 353
War Driving 353
Tools 354
NetStumbler 354
StumbVerter 354
DStumbler 355
Kismet 355
GPSMap 356
AiroPeek NX 357

AirSnort 357
WEPCrack 357
Detecting Wireless Attacks 357
Unprotected WLANs 357
DoS Attacks 358
Rogue Access Points 358
MAC Address Spoofing 358
Unallocated MAC Addresses 359
Preventing Wireless Attacks 359
Preventing Man-in-the-Middle Attacks 361
Establishing and Enforcing Standards for Wireless Networking 362
Case Study 362
Summary 365

Chapter 12 Using Trojans and Backdoor Applications 367
Trojans, Viruses, and Backdoor Applications 367
Common Viruses and Worms 368
Chernobyl 369
I Love You 370
Melissa 371
BugBear 372
MyDoom 373
W32/Klez 373
Blaster 375
SQL Slammer 376
Sasser 377
Trojans and Backdoors 378
Back Orifice 2000 378
Tini 389
Donald Dick 390
Rootkit 395
NetCat 395
SubSeven 398
Brown Orifice 411
Beast 412
Beast Server Settings 412
Beast Client 417
Detecting Trojans and Backdoor Applications 423
MD5 Checksums 424
Monitoring Ports Locally 424

Netstat 426
fport 428
TCPView 429
Monitoring Ports Remotely 429
Anti-virus and Trojan Scanners Software 430
Intrusion Detection Systems 431
Prevention 432
Case Study 433
Summary 436

Chapter 13 Penetrating UNIX, Microsoft, and Novell Servers 439
General Scanners 440
Nessus 440
SAINT 441
SARA 442
ISS 444
NetRecon 445
UNIX Permissions and Root Access 445
Elevation Techniques 446
Stack Smashing Exploit 446
rpc.statd Exploit 446
irix-login.c 447
Rootkits 447
Linux Rootkit IV 447
Beastkit 448
Microsoft Security Models and Exploits 448
Elevation Techniques 449
PipeUpAdmin 449
HK 449
Rootkits 450
Novell Server Permissions and Vulnerabilities 450
Pandora 451
NovelFFS 451
Detecting Server Attacks 452
Preventing Server Attacks 452
Case Study 456
Summary 458

Chapter 14 Understanding and Attempting Buffer Overflows 461

Memory Architecture 461

Stacks 461

Heaps 464

NOPs 464

Buffer Overflow Examples 465

Simple Example 465

Linux Privilege Escalation 466

Windows Privilege Escalation 471

Preventing Buffer Overflows 473

Library Tools to Prevent Buffer Overflows 475

Compiler-Based Solutions to Prevent Buffer Overflows 475

Using a Non-Executable Stack to Prevent Buffer Overflows 475

Case Study 476

Summary 479

Chapter 15 Denial-of-Service Attacks 481

Types of DoS Attacks 483

Ping of Death 483

Smurf and Fraggle 484

LAND Attack 485

SYN Flood 486

Tools for Executing DoS Attacks 486

Datapool 486

Jolt2 488

Hgod 489

Other Tools 489

Detecting DoS Attacks 489

Appliance Firewalls 490

Host-Based IDS 490

Signature-Based Network IDS 490

Network Anomaly Detectors 493

Preventing DoS Attacks 493

Hardening 494

Network Hardening 494

Application Hardening 496

Intrusion Detection Systems 497

Case Study 497

Summary 499

Chapter 16 Case Study: A Methodical Step-By-Step Penetration Test 501

Case Study: LCN Gets Tested 503

Planning the Attack 503

Gathering Information 504

Scanning and Enumeration 510

External Scanning 511

Wireless Scanning 511

Gaining Access 512

Gaining Access via the Website 512

Gaining Access via Wireless 522

Maintain Access 524

Covering Tracks 524

Writing the Report 524

DAWN Security 525

Executive Summary 525

Objective 525

Methodology 525

Findings 526

Summary 526

Graphical Summary 526

Technical Testing Report 527

Black-Box Testing 527

Presenting and Planning the Follow-Up 530

Part III Appendixes 533

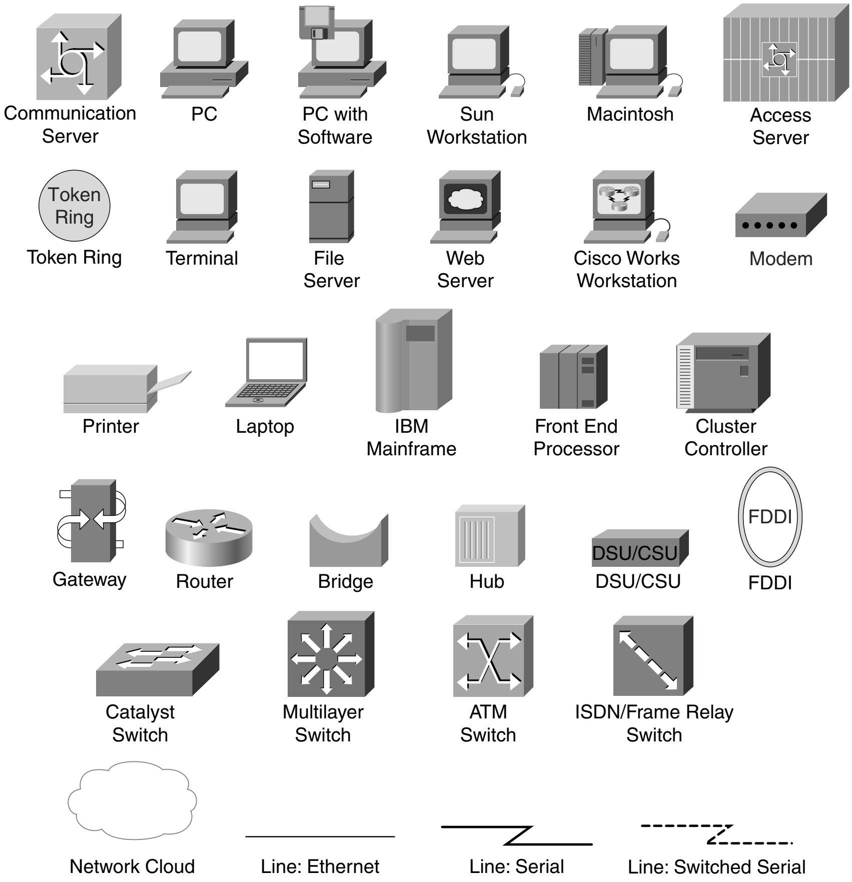
Appendix A Preparing a Security Policy 535

Appendix B Tools 547

Glossary 571

Index 583

Icons Used in This Book



Command Syntax Conventions

The conventions used to present command syntax in this book are the same conventions used in the IOS Command Reference. The Command Reference describes these conventions as follows:

- **Boldface** indicates commands and keywords that you enter literally as shown. In actual configuration examples and output (not general command syntax), boldface indicates commands that are manually input by the user (such as a **show** command).
- *Italic* indicates arguments for which you supply actual values.
- Vertical bars (|) separate alternative, mutually exclusive elements.
- Square brackets [ ] indicate optional elements.
- Braces { } indicate a required choice.
- Braces within brackets [{ }] indicate a required choice within an optional element.

Foreword

Pen testing, ethical hacking, posture assessment, vulnerability scans… the list of names goes on and on. There are as many names for simulating an attack and testing the security of an information system as there are approaches and techniques to be utilized in this endeavor.

While it is quite simple to log onto the web and gain access to tools, information, scripts, etc. to perform these types of tests, the key to doing this work responsibly, and with desirable results, lies in understanding how to execute a pen test the right way. Case studies have shown that a testing exercise designed to identify and improve security measures can turn sour and result in obvious or inaccurate recommendations, or in the worst case scenario, become disruptive to business operations.

This book goes to great lengths to explain the various testing approaches that are used today and gives excellent insight into how a responsible penetration testing specialist executes his trade.

Penetration testing is very dynamic field and requires a continuous investment in education and training to ensure that the tester has the requisite knowledge to do this well. And there is a certain elegance to the analysis involved in a truly successful test. While considered a science steeped in the world of technology, the highest form of penetration testing contains quite a lot of art. By applying creativity in the interpreting and analysis of results, then determining the optimal next steps, often by intuition and feel, the sophisticated pen tester creates a new level of evaluation and brings a stronger, more valuable result to the exercise.

There was a time 10-15 years ago when this type of exercise was questioned as to its validity, its value, and its interpretation. In today's modern technology-driven world, where we experience a ceaseless number of threats, vulnerabilities, DDOS attacks, and malicious code proliferation, penetration tests are one of many standard best practices essential to strong security governance. Most sound security approaches highlight these tests as an integral component of their programs. They are viewed as essential to understanding, evaluating, measuring, and then most importantly, establishing a cost effective set of remediation steps for improving the security of information assets.

What is of particular note and interest in this book is the extensive time devoted to the many new and innovative techniques required to properly test and evaluate new advanced technologies. It's an ever changing field and you will find great value in delving into these new domains, expanding your scope, and understanding the possibilities. There does not seem to be any limit to the potential damage that those with malicious intent can invoke. Deep exploration of their techniques helps us to establish proactive preventive and detective measures – and help in the ongoing tasks of staying a step ahead.

So when you do become involved in penetration testing projects – whether that be in contracting for services, overseeing their execution, reviewing their results, or even executing them yourself – it is essential to understand the concepts described within to ensure you have an evolved and sophisticated view of the world of penetration testing. Or was that ethical hacking?

Bruce Murphy
Vice President, World Wide Security Services
Cisco Systems, Inc.
September 2005

Introduction

The first “hackers” emerged from the Massachusetts Institute of Technology (MIT) in 1969. The term originally described members of a model train group who would “hack” the electric trains to increase the speed of their trains.

Today, the term has quite a different meaning. When people think of computer hackers, they think of computer experts who are adept at reverse engineering computer systems. They might think of malicious hackers who aspire to break into networks to destroy or steal data, or of ethical hackers who are hired to test the security of a network. Often, these ethical hackers, or penetration testers, mimic the same techniques as a malicious hacker.

The need for penetration testing is simple. The best way to stop a criminal is to think the way a criminal thinks. It is not enough to install burglar alarms and fences and assume that you are safe from burglary; to effectively stop a burglar, you must predict the moves a burglar would make. Likewise, to prevent against malicious hackers, you must think like a malicious hacker. One of the best ways that companies are assessing their security against attacks is by hiring outside security firms to attempt to penetrate their networks.

Companies are no longer falling victim to the “Titanic” syndrome. When the Titanic was built, its engineers never thought the ship would sink; companies now realize that just because their staff stamps their approval that the network is secure, you just do not know for sure until it is tested.

This book arises out of this need to know how to perform a thorough and accurate assessment of the network security for an organization. Although other books describe some of the tools that malicious hackers use, no book offered a definitive resource for penetration testers to know how to perform a full security assessment of a computer network for an organization. This book is written to fill this need.

Who Should Read this Book

The scope of this book is to provide a guide for those who are involved in the field of penetration testing, and for security professionals who daily face the need to know how to detect and protect against network attacks. It is specifically targeted toward three audiences:

- Those interested in hiring penetration testers
- Those employed as penetration testers
- Those responsible for securing their network against malicious hackers

Ethical Considerations

It should be noted at the onset that this book is designed as a guidebook for ethical hacking. This book does not endorse unethical or malicious use of the tools and techniques mentioned. Many of the techniques described in this book are illegal without prior written consent from an organization. The authors of this book want you to curb any curiosity you might have to try out these techniques on live systems without legitimate and ethical reasons. Used properly, the tools and techniques described in this book are an excellent resource for anyone who is involved in securing networks.

How This Book Is Organized

This book aids you in securing your network by examining the methods of penetration testing as a means of assessing the network of an organization. It also shows how to detect an attack on a network so that security professionals can spot an intruder and react accordingly. This book offers suggestions on how to go about protecting against the exploits discussed in each chapter. Numerous case studies are included throughout the book, and a complete case study chapter outlines a step-by-step example of the entire process.

This book is divided into three parts:

- Part I: Overview of Penetration Testing

  Before you can begin penetration testing, you must first comprehend the definition, purpose, and process of penetration testing. The first three chapters are devoted to meeting this objective.

  — Chapter 1: Understanding Penetration Testing

  This introductory chapter defines the scope and purpose behind penetration testing. Through the numerous examples of real-world security breaches coupled with statistics on the rise of security concerns, you learn the urgent need for this type of testing.

  — Chapter 2: Legal and Ethical Considerations

  Here you learn of the ethics, laws, and liability issues revolving around penetration testing. Mimicking the behavior of an attacker is a dangerous assignment; testers should understand what is permissible so that they do not step over the boundaries into unethical or illegal behavior.

  — Chapter 3: Creating a Testing Plan

  Because penetration testing requires such caution, it is imperative that the tester develop a step-by-step plan so that he can stay within his contracted boundaries. This chapter outlines the basic steps in performing a penetration test, which is further explained throughout the remainder of this book. Chapter 3 culminates with documentation guidelines for writing a synopsis report.

- Part II: Performing the Test

  The second part of this book focuses on the particulars of testing. Because the purpose of penetration testing is ultimately to assist administrators in securing their network, chapters include three essential components. First, the steps are given to perform a simulated attack using popular commercial and open-source applications. Only through a live test can one assess whether company security measures are effective. Second, when applicable, each chapter illustrates how to detect the attack through the use of the Cisco Intrusion Detection Sensor. Finally, each chapter concludes with some brief suggestions on how to go about hardening a system against attacks. All three components are essential in grasping the methods behind security breaches and how to prevent them from happening.

  — Chapter 4: Performing Social Engineering

  Social engineering is a component of testing that is often overlooked. It is the human element of the security assessment. Topics in this chapter include impersonations of technical support representatives, third-party companies, and e-mail messages.

— Chapter 5: Performing Host Reconnaissance

Host reconnaissance is the stake-out portion of testing. Often, a burglar patrols a street for several nights before his crime to determine which house might be the easiest to burglarize. During his stake-out, he examines each house closely, peeking in the windows. He is watching the behavior of its residents and evaluating the worth of goods inside. In the same way, a hacker performs reconnaissance to discover the hosts on a network and what applications and services are running.

In this chapter, you learn various reconnaissance techniques and software tools, besides how to spot and prevent a scan from being done on a network using the Cisco Intrusion Detection Sensor.

— Chapter 6: Understanding and Attempting Session Hijacking

In some secure environments, employees must swipe a card into a reader before being admitted through a door into their building. Although an intruder could certainly attempt to break in via a window, it would be easier to walk directly behind another employee as she walks into the building, thus bypassing its security.

Computer hacking has a similar technique called session hijacking. Here, a hacker monitors the traffic on a network and attempts to hijack a session taking place between a host and a server. By impersonating the identity of the host, the hacker is able to take over the session. As far as the server knows, it is still an authorized user accessing its services.

This chapter details the various methods that an attacker would use to hijack a session and how to detect and prevent session hijacking on a network.

— Chapter 7: Performing Web-Server Attacks

Nowadays it is rare for a company not to have some type of web presence. Whether it is just a simple static web page or a complex e-commerce site, companies know that if they want to compete in the market today, they must be accessible on the World Wide Web. Such a presence comes at a cost, however, because it leaves a potential opening for an attacker to enter a network of a corporation. Even if a malicious hacker cannot penetrate past the web server, he might be able to deface the website. If a customer sees that the website has been hacked, he might decide that he cannot trust the security of the company and take his business elsewhere.

This chapter walks you through exploiting web server vulnerabilities and how to detect and prevent against such attacks.

— Chapter 8: Performing Database Attacks

Before the age of computers, company files were often stored in locked file cabinets. Now they are stored in electronic databases. Unlike a locked file cabinet, however, a database is often not protected against curious intruders. Many times, databases are built with little or no security. The aim of this chapter is to show how to detect an attempt to breach database security through intrusion detection systems. It also instructs you on how to test the vulnerability of a database by emulating an intruder.

— Chapter 9: Cracking Passwords

Face it: Passwords are everywhere. You have to remember passwords for voice mail, e-mail, Internet access, corporate access, VPN access, and ATMs. With the number of passwords users have to remember, it is no wonder that they choose simple passwords and use the same one for multiple purposes. When users make the passwords simple, though, crackers (people who cracks passwords) can guess them easily through password-cracking tools. When users employ passwords repeatedly, if a cracker is able to crack one password, he then has access to all the services using the same password.

By the end of this chapter, you will know how to use some of the more popular password crackers to assess any easily guessed passwords on a network. You also will learn how to spot the signs of someone performing password cracking, and methods to prevent against it.

— Chapter 10: Attacking the Network

Historically, malicious hackers went after hosts on a network. Nowadays, the network itself can be a target, too. You can circumvent intrusion detection systems (IDSs), penetrate and bypass firewalls, and disrupt the service of switches and routers. This chapter covers these topics and provides a detailed examination of how to protect against such attacks through Cisco technology and proper network design.

— Chapter 11: Scanning and Penetrating Wireless Networks

Wireless networks are being implemented at a faster pace than ever before. The ease of being able to take your computer anywhere in an office building is attractive to most people, except, of course, the one in charge of IT security. Wireless networks, if not protected adequately, pose significant security threats. To secure a wireless network, an administrator should know the process by which an attacker would breach a wireless network, how to detect breaches, and how to prevent them. This chapter covers these topics.

— Chapter 12: Using Trojans and Backdoor Applications

It seems like every month, a new virus comes out. Virus protection software companies make a fortune in helping users protect against lethal viruses. Yet how do these viruses actually work? How do they enter a network? This chapter discusses Trojan horses, viruses, and other backdoor applications from the angle of a penetration tester who tries to mimic an attacker. It also points out preventative measures and how to detect suspicious behavior on a network that might reflect the existence of these malware programs on a network.

— Chapter 13: Penetrating UNIX, Microsoft, and Novell Servers

Administrators are fighting a never-ending war over which operating system is the most secure. Yet the inherent security in a default installation of popular server operating systems is not the real concern; the real concern is educating administrators on how to breach such operating systems. This chapter aids in this cause, taking a neutral stance among vendors and educating its readers in how to test their servers for vulnerabilities and protect against intruders.

— Chapter 14: Understanding and Attempting Buffer Overflows

A cargo ship only has so much capacity. If you have more items to transport than your cargo ship can handle, you may exceed its weight capacity and sink the ship. A buffer stack overflow operates in the same way. If an attacker is able to exceed the buffer's allocated memory, the application will crash. This chapter explains what a buffer overflow is, how to cause them, and methods for preventing them.

- Chapter 15: Denial-of-Service Attacks

  An attacker does not always want to read or alter confidential information. Sometimes an attacker wants to limit the availability of a host or network. He commonly does this through denial-of-service (DoS) attacks. This chapter describes some of the more common methods of performing such attacks, how to detect them, and how to prevent them.

- Chapter 16: Case Study: A Methodical Step-By-Step Penetration Test Example

  Using a mock organization, this concluding chapter outlines the steps that a penetration tester takes as he performs reconnaissance, gains access, maintains that access, and captures valuable intellectual property. The fictitious tester then covers his tracks by erasing logs to prevent detection.

- Part III: Appendixes

  The final part of this book includes supplementary material that covers the next step to take after completing a penetration test.

  - Appendix A: Preparing a Security Policy

    Any security weaknesses discovered during testing are not a reflection on poor technology, but on weak security policies. This appendix provides a basic example of a security template that you can use as a template for developing your own policy.

  - Appendix B: Tools

    Every ethical hacker has a favorite software “toolkit” containing his preferred applications used in testing or auditing. Numerous commercial and noncommercial software tools are mentioned throughout this book. This appendix consolidates all descriptions of the prominent tools in one easy location. Each tool is referenced alphabetically by chapter and contains a website reference for the software. You can also find a hyperlinked PDF version of this appendix at http://www.ciscopress.com/title/1587052083 to easily launch your web browser to the URLs listed.

  - Glossary

    The glossary defines a helpful list of terms used commonly in various facets of penetration testing practice.

We believe you will find this book an enjoyable and informative read and a valuable resource. With the knowledge you gain from studying this book, you will be better fit to secure your network against malicious hackers and provide a safer place for everyone to work.

Failing to prepare is preparing to fail.

—John Wooden (Former head coach, UCLA men's basketball team)

CHAPTER 3

Creating a Test Plan

As with all great projects, success comes with having a solid methodical plan. Penetration testing is not about jumping into a security assessment project by running several tools at random. Penetration testing is about creating a methodical, step-by-step plan that details exactly what you are going to do, when you are going to do it, and how.

This chapter outlines the steps needed to create a methodical plan, from narrowing the scope of the project, to using the Open-Source Security Testing Methodology Manual (OSSTMM), and finally to writing up the testing report.

Step-by-Step Plan

Every good penetration test involves the following steps:

1. **Reconnaissance**—The initial stage of collecting information on your target network
2. **Enumeration**—The process of querying active systems to grab information on network shares, users, groups, and specific applications
3. **Gaining access**—The actual penetration
4. **Maintaining access**—Allowing the tester a backdoor into the exploited system for future attacks
5. **Covering tracks**—The process of deleting log file entries to make it appear that you were never on the exploited system

Chapter 5, “Performing Host Reconnaissance,” addresses the reconnaissance step. The last four steps, which are typically done in sequence, are covered in the remaining chapters.

Before you can perform the first step, however, you and the client (or management, if you are doing an internal test) must do the following:

- Narrow the scope of the project
- Determine if social engineering will be employed
- Decide if session hijacking attempts will be allowed
- Agree on the use of Trojan and backdoor software

Defining the Scope

Penetration testing is a lot like a pirate looking for buried treasure. The pirate does not know exactly where the buried treasure is, but he knows it is valuable enough to go looking for it. A pirate has a treasure map full of clues all geared to direct him toward the buried treasure. In the same way, penetration testers are on a quest to infiltrate a client network. The testers do not know in advance how they are to go about infiltrating the network, but in the end, the results of the test have to be worthwhile to the client. If a client is most concerned with the security of their Internet presence, then you should not devote your time to trying to break into the internal network. Likewise, if the client is concerned only about the security of his accounting department, it does not make sense to devote your time to other departments.

The first step, then, is to narrow the scope of your test to what is meaningful to the client. Ask the client what he hopes to achieve through this testing. Perhaps he only wants to assess whether he is vulnerable to having account information stolen, or the scope might extend to any type of attack. Ideally, all possible means of attacks should be allowed to provide the most realistic scenario of a real malicious attack, but this is seldom the case. Budget constraints, concerns over denial of service (DoS) attacks disrupting daily information, and the protection of employee privacy are often deterrents that prevent organizations from authorizing all forms of attacks.

Social Engineering

Social engineering, described in more detail in Chapter 4, "Performing Social Engineering," is the process of human-based manipulation to achieve access. Some organizations permit the use of social engineering, and some do not. You need to discuss this with the client (and have it in writing) before you begin testing.

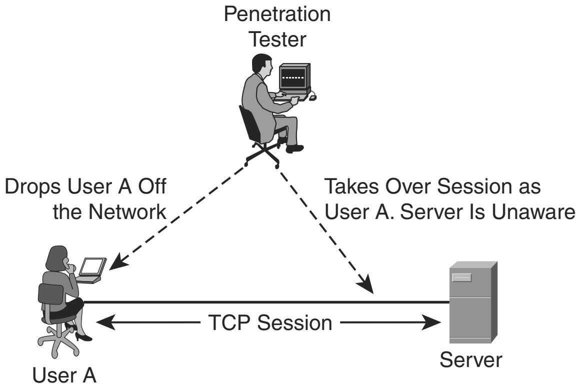
Session Hijacking

Session hijacking, described in more detail in Chapter 6, "Understanding and Attempting Session Hijacking," is the process of taking over a TCP session between two machines to gain access to an unauthorized system, as illustrated in Figure 3-1.

In Figure 3-1, the penetration tester is listening to network traffic being sent from User A to the server. The penetration tester takes over the session and appears to the server as that user. To make this work, the penetration tester has to drop User A off the network (usually through sending a TCP reset packet). This can be disruptive to day-to-day operations and it is often not permissible to perform these tests.

An alternative is to create a lab environment that contains equivalent network equipment.

Figure 3-1 *Session Hijacking*


Penetration Tester
Drops User A Off the Network
Takes Over Session as User A. Server Is Unaware
TCP Session
User A
Server

Trojan/Backdoor

Another factor requiring authorization before performing tests is whether the use of Trojans or other backdoor software is to be allowed. Encourage the client to allow this. Many of the more cunning attacks use backdoor applications and Trojans. If you want to have accurate results, you need authorization to use these applications.

If you do agree on the use of Trojan applications and other backdoor applications, be careful about what tools you use. Some websites give you the option of downloading Trojan and backdoor tools such as Netcat, but they contain their own virus embedded in the program. These viruses, when put on a client machine, can propagate throughout the network, causing havoc on servers and end user computers.

Open-Source Security Testing Methodology Manual

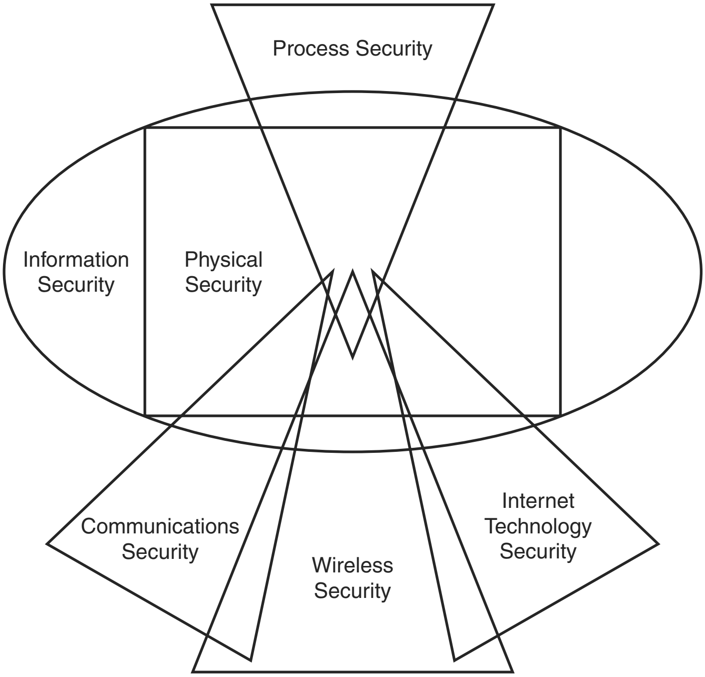
As you know, it is pointless to reinvent the wheel if it has already been made. Peter Herzog, at the Institute for Security and Open Methodologies (http://www.isecom.org), along with 30 contributors from various security organizations, has created the Open-Source Security Testing Methodology Manual (OSSTMM) so that penetration testers do not have to reinvent the wheel when designing a methodology for security auditing.

The OSSTMM addresses the following areas of security assessment, as illustrated in Figure 3-2:

- Information security
- Process security
- Internet technology security
- Communications security

- Wireless security
- Physical security

Figure 3-2 *OSSTMM Security Map*



©2000–2003 Peter Herzog, ISECOM

NOTE A Spanish version of the OSSTMM is available for free download at http://www.osstmm.org.

Each of the areas of security assessment is further broken down into specific modules. For example, the wireless security area (page 71 in the OSSTMM document) is broken down into eleven modules:

- Electromagnetic radiation testing
- 802.11 wireless network testing
- Bluetooth testing
- Wireless input device testing

- Wireless handheld testing
- Cordless communications testing
- Wireless surveillance device testing
- Wireless transaction device testing
- RFID testing
- Infrared testing
- Privacy review

Each of these modules is further broken down to detail what a security auditor should test. For example, under Bluetooth testing (page 75), the auditor should do the following:

1. Verify that there is an organizational security policy that addresses the use of wireless technology, including Bluetooth technology.

2. Perform a complete inventory of all Bluetooth wireless devices.

3. Perform brute force attacks against Bluetooth access points to discern the strength of the password. Verify that passwords contain numbers and special characters. Bluetooth access points use case-insensitive passwords, which makes it easier for attackers to conduct a brute force guessing attack due to the smaller space of possible passwords.

4. Verify the actual perimeter of the Bluetooth network.

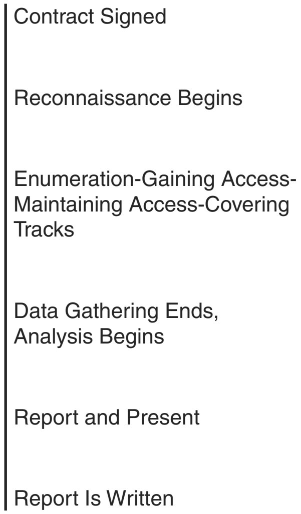
5. Verify that the Bluetooth devices are set to the lowest power setting to maintain sufficient operation that will keep transmissions within the secure boundaries of the organization.

The OSSTMM, although broader than just penetration testing, serves as a good framework to start with.

NOTE Anyone can contribute to the OSSTMM project. If you want to contribute to it, go to http://www.isecom.org/contact.shtml.

After you have collected the data, you can begin your assessment. Figure 3-3 illustrates the complete process from the point of signing the contract to the point of writing the report.

Figure 3-3 *Penetration Testing Life Cycle*


Contract Signed
Reconnaissance Begins
Enumeration-Gaining Access-Maintaining Access-Covering Tracks
Data Gathering Ends, Analysis Begins
Report and Present
Report Is Written

After you have collated and analyzed all data, it is time to write your report.

Documentation

A penetration test is useless without something tangible to give to a client or executive officer. A report should detail the outcome of the test and, if you are making recommendations, document the recommendations to secure any high-risk systems.

The report should contain the following sections:

- Executive Summary
- Project Scope
- Results Analysis
- Summary
- Appendixes

Executive Summary

The Executive Summary is a short high-level overview of the test. It is written for key executives who want to know the bottom line about how this affects their company but probably do not care much about the technical details. A sample Executive Summary would read as follows:

Executive Summary

This report details a recent intrusion test on <client name> as performed by <testing firm> between the dates of <dates>. <Client> contracted <testing firm> on <date of signed contract> to assess the security of <client>'s [public/private] network by emulating the techniques of a malicious attacker. A combination of tests was executed against <client name> [public/private] network, including port scans, exploit tests, ICMP scans, and other means to be detailed later in the report.

After reviewing the results of the tests, <testing firm> recommends the following to improve network security:

<bulleted list of suggestions>

Included in this report is a brief introduction about intrusion testing and an explanation of the scope of tests performed. This is followed by the complete results of the test and assessments of the results.

As the sample demonstrates, you should keep the Executive Summary brief. It is usually only a page long. You might encounter executive officers who stay only long enough for a brief five-minute introduction and overview of the Executive Summary followed by a question and answer period. Therefore, you should keep your Executive Summary brief and to the point within the context of how the results impact the business as a whole.

Your Executive Summary should also include a business case detailing the impact of your findings and any associated costs in fixing discovered vulnerabilities. You can use charts to support your case and make the report easier to read.

As a penetration tester, you are considered a specialist. You are hired to give not just your findings but also an analysis. You should include in your Executive Summary information on how your client compares with other companies you have performed tests on. To preserve confidentiality, you should not offer the names of any other clients, but instead provide generic statements as to whether the security of the company falls short or excels when compared to other companies in the same industry.

TIP Because some of the officers might be unfamiliar with the need or purpose of penetration testing, the best practice is to include a one-page description after the Executive Summary explaining why penetration testing is important and what it entails. Include statistics and define common terms that you will use throughout the remainder of the report. This piques the interest of the readers and illustrates the importance of your work.

Project Scope

The Project Scope should include the IP address range tested against and the boundaries defined in the contract. The boundaries include such things as whether you employed social engineering, whether you tested the public (Internet-facing) or private networks, and whether you permitted Trojans and backdoor software applications such as Back Orifice. Although the timeframe for the test is included in the Executive Summary, you should include it here, too, because it relates to the Project Scope.

You should also include an estimate of the number of exploits attempted and their type. For example, the report might say this:

> More than 230 tests were performed against hosts. These included, but were not limited to, the following:
>
> - Backdoor application vulnerabilities
> - CGI vulnerabilities
> - FTP server vulnerabilities
> - Game server vulnerabilities
> - Mail server vulnerabilities
> - Other server vulnerabilities
> - Network-based services vulnerabilities
> - Firewall vulnerabilities
> - Remote administration vulnerabilities
> - Web server vulnerabilities
> - CERT/CC advisory testing
> - BugTraq advisory testing
> - Dictionary attacks
> - CGI scanner
> - Port scanner
> - ICMP tests

Results Analysis

The Results Analysis is the meat of the report. The length of this section can vary from as few as ten pages to as many as several hundred pages, depending on the scope and detail of the tests. You should use a base template for this section, including the following:

- IP address and domain name of host
- Listening TCP and UDP ports

- Service description
- Tests performed
- Vulnerability analysis

The following is a sample results analysis.

IP: 172.16.22.199 Name: CorpWebSrvr1

Port	Service	Description
80	HTTP (Web)	Host appears to be running Microsoft Internet Information Server 5.0. Attempts to penetrate included the following: 1) msadc exploit, 2) codebrw.asp exploit, 3) showcode.asp exploit, 4) cgi exploits, 5) webhits.dll / webhits.htw exploits, 6) $data exploit, 7) ASP dot bug exploit, 8) ISM.dll buffer truncation exploit, 9) .idc and .ida exploits, 10) +htr exploits, 11) adsamples exploit, 12) /iisadmnpasswd, 13) dictionary password cracking, 14) brute force password cracking, and 15) SQL injection.
443	HTTPS (Secure Web)	A 1024-bit digital certificate is used that will expire December 15, 2005. The certificate is encrypted using RSA Sha1 encryption and is signed by VeriSign.

Vulnerability Analysis

Vulnerability: Unicode Directory Traversal
Risk: High
Description: A flaw in IIS allows for a malicious hacker to execute code on a target system. During testing, the following was entered into the URL string in a Microsoft Internet Explorer web browser:

```
http://www.hackmynetwork.com/scripts/..%co%af%../..%co%af%../..%co%af%../
  ..%co%af%../..%co%af%../..%co%af%../..%co%af%../..%co%af%../winnt/system32/
  cmd.exe?/c+dir+c:
```

This resulted in getting a complete directory listing of the target server. You can use this same syntax to execute code on a target system. Attackers can use this exploit to steal confidential information, launch another attack, or perform DoS attacks on the target network.

Vulnerability: IIS Sample Codebrws.asp
Risk: Medium
The codebrws.asp sample file is shipped with Microsoft IIS server and can be used to remotely read arbitrary files. This might reveal sensitive information or code that can be used for further exploits.

Summary

The Executive Summary at the beginning of the report is directed toward key decision makers; the final Summary is directed toward technical personnel. This section should contain a bulleted list of technical recommendations for the client.

Appendixes

Finally, your report should include appendixes that include the following:

- Contact information
- Screen shots
- Log output

Screen shots and log output are especially important. You should document everything you do during the test to prove your work to the client.

When you present your client with the report, he should sign a receipt for it to acknowledge that you have turned over your only copy of it and that you cannot be expected to reproduce copies of the report without doing the work again. Your report should be digitally signed and presented in a form that prevents editing, such as PDF files. The footer of each page should state that the information is confidential.

After you have presented your report, you need to agree with your client as to what to do with your copy of it. Recommended practice is to shred any hard copies you have and delete any soft copies using disk wiping software such as PGP.

Summary

This chapter presented an introduction to the process of creating a test plan for performing a penetration test. Penetration testing includes the following steps:

1. Reconnaissance
2. Enumeration
3. Gaining access
4. Maintaining access
5. Covering tracks

Before you get started, you should devise a methodical plan on how you are to perform your test. You can use the Open-Source Security Testing Methodology Manual (OSSTMM) as a starting guide.

After you finish the test, you construct a report. The report should contain each of the following:

- Executive Summary
- Project Scope
- Results Analysis
- Summary
- Appendixes

After you present the report, the next step is to discuss policies. Any vulnerability that exists on a network of an organization is either because the organization is not following its security policies or because an important component is missing from its security policy. You can read more about security policies in Appendix A, “Preparing a Security Policy.”

This page intentionally left blank

INDEX

Numerics

- **802.1x port security, 352**

A

- **access auditing, enabling, 266**
- **access points, 350**
- **account lockouts, 311**
  - detecting password-cracking attacks, 307
- **account logins, logging, 309–310**
- **ACK scan, discovering firewall configuration, 321**
- **ACK scans, 100**
- **ACK storms, 137–138**
- **acting classes, developing social engineering skills, 57**
- **active host reconnaissance, 89**
  - NSLookup/Whois lookups, 89–92
  - SamSpade, 92–94
  - Visual Route, 95
- **active session hijacking, 127**
- **AD (active directory) model, 448**
- **admin account (Novell), 451**
- **AiroPeek NX, 357**
- **AirSnort, 357**
- **Allaire, J.J., 193**
- **ALOHANET, 9**
- **anomaly detection systems, 109**
- **anomaly-based IDSs, evading, 324**
- **antennas, 350**
- **antivirus scanners, 430–431**
- **Apache HTTP Servers, 11**
  - securing, 236
  - vulnerabilities, 199
- **appendixes, 44**
- **appliance firewalls, detecting DoS attacks, 490**
- **application hardening, 496**
  - DoS attacks, preventing, 497
- **APs (access points), *detecting* rogue APs, 358**
- **ARP attacks**
  - hardening switches against, 341
  - testing switches for vulnerabilities, 335
- **ASA (Adaptive Security Algorithm), 337**
- **ASP, 188–190**
- **assets**
  - threats to, identifying, 537
  - cost of protecting, identifying, 537–538
- **assigning permissions to root user, 445**
- **Atkins, Steve, 93**
- **attacks**
  - D.A.D. attacks, 7
  - DoS, 21
  - hacktivist attacks, 13
  - mutating, 324
  - on databases, protecting against, 270–271
  - stages of
    - *erasing evidence, 14*
    - *maintaining access, 14*
    - *obtaining access, 14*
    - *reconnaissance stage, 13*
    - *scanning stage, 14*
  - zero-day, 8
- **auditing passwords, 309**
- **auditing tools**
  - AiroPeek NX, 357
  - AirSnort, 357
  - DStumbler, 355
  - GPSMAP, 356
  - Kismet, 355
  - NetStumbler, 354
  - StumbVerter, 354
  - WEPCrack, 357
- **authentication, 210**
  - on Microsoft SQL Server, sysxlogins, 261–262
- **authority-based persuasion, 53**
- **authorship of security policies, 540**
- **availability of hacking tools, 10**
- **availability threats, 7**

B

- **backdoor applications, 37, 560, 562**
  - detecting, 423
  - preventing, 432–433
- **backup policies, 543**
- **bandwidth attacks, 481**
- **banner grabbing, 223**
- **basic authentication bypass attacks, 199**
- **basic HTTP authentication, 210**
- **Beast, 412**
  - client configuration, 417–419, 423
  - gaining access with, case study, 433–436
  - server settings, 412–416

Beastkit, 448
behavioral profiling, 55
behavior-based virus detection, CSA, 456
Berners-Lee, Tim, 180
black-box tests, 6
black-hat hackers, 5
Blaster worm, 375–376
blind hijacking, 130–131
blind spoofing, 129
BO2K (Back Orifice 2000), 378–384, 388
 plug-ins, 380–382
Borghard, Peter, 27
BoSniffer program, 67
Boson GetPass, 302–303
Boson RIP Route Generator, 330
breaching internal networks, case study, 238–244
Brown Orifice, 411–412
Brumleve, Dan, 411
brute force attacks, 209
 Brutus, 211
 detecting, 212, 215
 HTTP Brute Forcer, 211
 on SQL servers
  case study, 272, 275–277
  detecting, 268
 protecting against, 215–216
 testing database vulnerability to, 260
Brutus, 211
buffer overflows
 case study, 476–478
 heap buffers, 464
 Linux privilege escalation, 466–471
 on IIS web servers, 202–203
 preventing, 473–475
 simple buffer overflow example, 465
 stack smashing exploit, 446
 stacks, 461
 tools for attempting, 566
 Windows privilege escalation, 471
BugBear virus, 372–373
bypassing firewalls, 322
 ICMP tunneling, 323
 Loki ICMP tunneling, 322
bytecode, 466

C

calculating cost of protecting assets, 537–538
candy networks, 49
case studies
 buffer overflows, 476–478
 DoS attacks, 497–499
 gaining access with Beast, 433–436
 internal networks, breaching, 238–244
 password cracking, 316–318
 port scanning, 118, 121–122
 remotely accessing networks, 342, 345–347
 rootkits, 456–457
 session hijacking, 168–169, 172–173
 SQL server brute force attacks, 272, 275–277
 wireless network security, 363–365
cDc (Cult of the Dead Cow), 378
CDP (Cisco Discovery Protocol)
 disabling on routers, 338
 testing for vulnerabilities, 324–326
CERT (Computer Emergency Response Team), 8
CFAA (Computer Fraud and Abuse Act), 25
CFML (ColdFusion Markup Language), 193
CGI (Common Gateway Interface), 191
character traits of social engineers
 confidence, 57–58
 patience, 56–57
 possessing inside knowledge, 59
 trust, 58
characteristics of open source software, 11
Chernobyl virus, 369
CIA (confidentiality, integrity and availability), 7
Cisco Guard XT, 493
Cisco IDS 4200 series, detecting session hijacking, 153–156, 160–164
Cisco passwords, cracking, 302–303
Cisco PIX Firewalls
 Flood Defender, 490
 securing, 337
Cisco Traffic Anomaly Detector XT appliance, 493
CleanIISLog, 222
clear-text passwords, 280
client-based Java, 194
Code of Fair Information Practices, 25
ColdFusion, 193
commands, SQL, 249
commercial vulnerability scanners, 219
company websites, downloading for offline viewing, 79–80
compiler-based solutions, preventing buffer overflows, 475

compiling NMAP, 105
Computer Ethics Institute, Ten Commandments of Computer Ethics, 22–23
Computer Misuse Act of 1990 (U.K.), 24, 31
confidence as social engineer trait, 57–58
confidentiality threats, 7
configuring
- Ethereal, 145
- password settings, 311–312

conformity persuasion, 50
connection strings, testing vulnerability to, 259
connnections, embryonic, 140
conscientious behavior classification (D.I.S.C.), 55
cookie guessing, 205–206
cracking encrypted passwords, 281
- tools, 556

creating
- security policies, 544
- spoofed e-mail messages, 65–67

crystal-box tests, 6
CSA (Cisco Security Agent), 455
customer impersonation, 69–70
cybercrime, 23

D

D.A.D. (disclosure, alteration, and destruction) attacks, 7
D.I.S.C. profile, 55
data dictionary, 250
database attacks
- detecting, 266
- tools for performing, 556

databases
- Microsoft SQL Server, 252
  - *querying, 253*
  - *structure of, 252*
- MySQL, 251
- Oracle, 250
- protecting against attacks, 270–271
- RDBMSs, 249
  - *vulnerabilities, testing, 254–260*
- susceptibility to attacks, 249

Datapool, 486
DAWN Security Systems, fictitious penetration test of LCN, 505
- covering tracks, 524
- follow-up, 530
- gaining access, 512–517, 520, 523
- information gathering, 504, 507–510
- maintaining access, 524
- planning the attack, 503
- sample report, 524–529
- scanning and enumeration, 510–512

DDoS (Distributed Denial of Service) attacks, 482
DDoSPing tool, 492
default SSIDs as security risk, 351
defending against social engineering attacks, 71
defining scope of testing, 6, 36
detecting
- backdoor applications, 423
- brute force attacks, 212, 215
- database attacks, 266
- directory traversals, 226–228
- DoS attacks, 489
  - *with appliance firewalls, 490*
  - *with host-based IDSs, 490*
  - *with signature-based network IDSs, 490, 492*
- password-cracking attacks, 305–307
  - *from physical access, 308*
  - *within system log files, 306*
- scans, 109
- server attacks, 452
- session hijacking, 143–144
  - *with Cisco IDS 4200 series sensors, 153–156, 160–164*
  - *with packet sniffers, 145*
- Trojans applications, 423
- web attacks, 225
  - *directory traversals, 226–228*
- Whisker attacks, 228–231
- wireless network attacks
  - *DoS attacks, 358*
  - *MAC address spoofing, 358*
  - *on unprotected WLANs, 357*
  - *rogue APs, 358*
  - *unallocated MAC addresses, 359*

devices, 324, 333
- routers, testing for vulnerabilities, 324–332
- switches, 111–112
  - *testing for vulnerabilities, 333–337*

DHTML (Dynamic HTML), 181
DilDog, 378
dipole antennas, 350

directory traversals, detecting, 226–228

disabling unnecessary router services, 338

disaster recovery policies, 543

discovering firewall configurations

- through firewalking, 321
- with ACK scan, 321

DMZ (demilitarized zone), securing network architecture, 237

documentation

- Appendixes, 44
- Executive Summary, 40, 44
  - *sample, 41*
- Project Scope, 42
- Results Analysis, 42–43

domain model, 448

dominant behavior classification (D.I.S.C.), 55

Donald Dick, 390–393

DoS (denial of service) attacks, 21, 481

- case study, 497–499
- DDoS, 482
- detecting, 489
  - *with appliance firewalls, 490*
  - *with host-based IDSs, 490*
  - *with signature-based network IDSs, 490–492*
- detecting on wireless networks, 358
- Fraggle attacks, 485
- LAND attacks, 485
- Ping of Death, 483–484
- preventing, 493
  - *through application hardening, 497*
  - *through network hardening, 494–496*
- protocol exceptions, 481
- Smurf attacks, 484
- SYN floods, 486
- tools for executing, 486, 566–569
  - *Datapool, 486*
  - *Hgod, 489*
  - *Jolt2, 488*

downloading websites for offline viewing, 79

DSSS (Direct Sequence Spread Spectrum), 350

DStumbler, 355

DTD (Document Type Definition), 183–184

dumb scans, 101

dumpster diving, 308

- obtaining company information, 59

E

EAPOW (Extensible Authentication Protocol over Wireless), 352

e-commerce, 198

EDGAR (Electronic Data Gathering, Analysis, and Retrieval), 78, 87

educating employees about password cracking, 315–316

EIP (extended instruction pointer), 202

elements (HTML), 180

elevation techniques, 446

- for Microsoft Windows, 449
- irix-login.c exploit, 447
- rpc.statd exploit, 446–447
- stack smashing exploit, 446

Elkern virus, 373

e-mail impersonation, 64–68

- malware attachments, 67

e-mail policies, 541

embryonic connections, 140

employees

- impersonating, 60
- tech support, impersonating, 61–62
- telecommuter phone numbers, obtaining, 61

enabling access auditing, 266

encrypted passwords, 281

end-user impersonation, 69

enforcing standards for wireless networks, 362

EPHI (electronic public health information), 29

erasing

- evidence stage of attacks, 14
- passwords, 280

ESP (extended stack pointer) registers, 462

Ethereal

- configuring, 145
- monitoring session hijacking attacks, 147, 151–153

ethical hackers, 5

European Council Convention on Cybercrime, 23

evading IDSs, 323

example penetration test, LCN, 501–517, 520, 523–530

examples

- of buffer overflows
  - *Linux privilege escalation, 466–471*
  - *Windows privilege escalation, 471*
- of scan detection, 112
  - *FIN scans, 115*

NULL scans, 115
OS guessing scans, 117–118
SYN scans, 114
TCP Connect() scans, 113
Xmas tree scans, 115
of simple buffer overflows, 465
execiis-win32.exe, 221
executing DoS attacks, 486
Datapool, 486
Hgod, 489
Jolt2, 488
Executive Summary, 40–41, 44
expanding number of security threats, reasons for
administrator trust, 13
availability of hacking tools, 10–11
business partnerships, 13
complexity of networks, 10
frequency of software updates, 10
hacktivism, 13
industry regulations, 12
marketing demands, 12
nature of open source, 11
proliferation of viruses and worms, 9
reliance on Internet, 11
unmonitored telecommuters and mobile users, 12
WLANs, 9
expiration dates of passwords, modifying, 312
exploits
hidden fields, 207–209
zero-day, 8
extended stored procedures, 258, 272
external JavaScript, 185
extra fun tab (SubSeven), 411

F

fictitious penetration test of LCN, 503, 505
covering tracks, 524
follow-up, 530
gaining access, 512–517, 520, 523
information gathering, 504, 507–510
maintaining access, 524
planning the attack, 503
sample report, 524–529
scanning and enumeration, 510, 512
file extensions for web languages, 178
file permissions, UNIX, 445
FIN scans, 100
fingerprinting, 106
firewalking, discovering firewall configuration, 321
firewalls
bypassing, 322
ICMP tunneling, 323
Loki ICMP tunneling, 322
discovering configuration through firewalking, 321
securing, 337
FISMA (Federal Information Security Management Act), 30
Flood Defender, 490
footprinting, 108, 501
form-based authentication, 216
form-based HTTP authentication, 210
four-factor security, 279
fport, 428
Fraggle attacks, 485
fun manager tab (SubSeven), 411
FuX0red, 398
fuzzy matching, 107

G

general-purpose registers, 462
GetPass, 302–303
GLB (Graham-Leach-Bliley) act of 1999, 29
GNU (GNUs not UNIX), 11
Google.com as hacking tool, 224
Gosling, James, 193
GPSMAP, 356
grading passwords with L0phtcrack, 295
gratuitous ARPs, 138
gray-box testing, 6, 31
guest accounts (Microsoft SQL Server), 264

H

hacking, 5
laws. *See* legislation
non-U.S. laws pertaining to, 31
Novell NetWare, 451
script-kiddies, 11
tools, availability of, 10
hacktivism, 13
hardening switches
against ARP attacks, 341
against MAC table flooding attacks, 341

against STP attacks, 341
against VLAN hopping, 341
against VTP attacks, 342
hashing algorithms, 281
Hassell, Riley, 202
heap buffers, 464
Hedley, David, 447
Herzog, Peter, 37
Hgod, 489
hidden field exploit, 207–209
Hide Server, 266
high-gain omni-directional antennas, 350
highly directional antennas, 350
HIPAA (Health Insurance Portability and Accountability Act), 12
history of passwords, modifying, 313
history of wireless networking, 349–350
HK exploit, 449
hoaxes, BugBear virus hoax, 373
honeypots, 17
host reconnaissance, tools, 547, 550–553
host-based IDSs, 110
detecting DoS attacks, 490
HTML, 179
HTTP
basic authentication, 210
form-based authentication, 210
tunneling, 323
web server vulnerabilities
Apache, 199
IIS, 199–203
HTTP Brute Forcer, 211
HTTP service
disabling on routers, 338
testing for vulnerabilities, 326
human psychology, behavioral profiling, 55
human-based social engineering, 50
customers, impersonating, 69–70
defending against, 71
e-mail impersonation, 64–68
employees, impersonating, 60
tech support, 61–62
end-users, impersonating, 69
persuasion, types of
authority-based, 53
conformity-based, 50
information-based, 54–55
logic-based, 51–52
need-based, 52
reciprocation-based, 53–54
similarity-based, 54
RSE, 70
third parties, impersonating, 62–64
traits of social engineers
confidence, 57–58
patience, 56–57
possessing inside knowledge, 59
trust, 58
wetware, 50
Hunt, 134–135
hijacking Telnet sessions, 135
HyperText PreProcessor, 192
Hypnopædia, 299

I

"I Love You" worm, 370
ICMP
Ping of Death attacks, 483–484
Smurf attacks, 484
tunneling, 322–323
identifying threats to assets, 537
IDSs (intrusion detection systems), 109, 431
evading, 323
IEV (IDS Event Viewer), viewing session hijacking, 164–165
IIS Lock Down, 235
IIS web servers
securing, 234–236
vulnerabilities, 199–200
buffer overflows, 202–203
privilege escalation, 201
showcode.asp, 200–201
IIS Xploit, 221
impersonating
company employees, 60
tech support personnel, 61–62
customers, 69–70
end-users, 69
third parties, 62–64
implementing
employee education on password cracking, 315–316
physical protection, 313–314
improvisation, developing social engineering skills, 57

industry regulations as cause for expanding number of security threats, 12
influential behavior classification (D.I.S.C.), 55
INFORMATION_SCHEMA views, 251
information-based persuasion, 54–55
inside knowledge, importance of possessing, 59
integrity threats, 7
IntelliTamper, 222
internal networks, breaching, case study, 238–244
Internet policies, 541
interpreted languages, Perl, 187
IPSec, deploying on wireless networks, 353
IPv6 URI parsing heap overflows, 199
irix-login.c exploit, 447
ISS (Internet Security Scanner), 444
Ivgi, Rafel, 208
I-worm, 373

J

Java, 193
- client-based, 194
- server-based, 194–195

JavaScript, 185–186
John the Ripper, 285–287
johnny.ihackstuff.com, 224
Jolt2, 488
JScript, 186
Juggernaut, 131–133

K-L

key logging, 308
Kismet, 355
Klez worm, 373
Kouznetsov, Pavel, 194
Krauz, Pavel, 134
Kwyjibo, 371

L0phtcrack, 289–290, 294, 298
LAND attack, 482
LANMAN hashing, 283
LCN (Little Company Network), example penetration test, 501–505
- covering tracks, 524
- follow-up, 530
- gaining access, 512–523
- information gathering, 504, 507–510
- maintaining access, 524
- planning the attack, 503
- sample report, 524–529
- scanning and enumeration, 510–512

legislation
- CFAA, 25
- Code of Fair Information Practices, 25
- Computer Misuse Act (U.K.), 24
- FISMA, 30
- Graham-Leach-Bliley act of 1999, 29
- non-U.S., 31
- Sarbanes-Oxley Act, 30
- USA PATRIOT Act, 30

length of passwords, modifying, 312
Lerdorf, Rasmus, 192
library tools, preventing buffer overflows, 475
Linux
- DoS attacks, executing
  - *Spendor Datapool, 486*
- elevation techniques
  - *irix-login.c exploit, 447*
  - *rpc.statd exploit, 446–447*
  - *stack smashing exploit, 446*
- MACOF, 129
- privilege escalation, example of, 466–471
- rootkits, 447

Litchfield, David, 260
LiveScript, 185
Local System account (SQL Server), 263
locking out user accounts, 311
logging
- account logins, 309–310
- actions during penetration testing, 31–32

logic attacks, 482
logic persuasion, 51–52
Loki ICMP tunneling, 322
Long, Johnny, 224

M

MAC address spoofing, detecting on wireless networks, 358
MAC filtering on wireless networks, 352
MAC table flooding attacks
- hardening switches against, 341
- testing switches for vulnerabilities, 335

MACOF, 129
maintaining access stage of attacks, 14
malware, 67, 368
markup languages
  CFML, 193
  DHTML, 181
  HTML, 181
  XHTML, 184
  XML, 183–184
master database, 252
MBSA (Microsoft Baseline Security Analyzer), 270
MD5 checksums, 424
Melissa virus, 371
memory, buffers
  heap buffers, 464
  stacks, 461
memory consumption DoS attacks, 199
Microsoft
  AD model, 448
  domain model, 448
  password hashing, 282
    LANMAN hashing, 283
    Windows hashing, 282
  SQL Server, 252
    authentication, sysxlogins, 261–262
    guest accounts, 264
    network libraries, 264–265
    ports, 265
    Public role, 263
    querying, 253
    sample databases, 264
    service accounts, 263
    structure of, 252
  Windows
    elevation techniques, 449
    rootkits, 450
mirror technique for gaining employee trust, 58
misuse detection systems, 109
MITM (man-in-the-middle) attacks, 128
  preventing on wireless networks, 361
  session hijacking
    blind spoofing, 129
    case study, 168–169, 172–173
    nonblind spoofing, 128
    TCP sequence prediction, 130–131
Mitnick, Kevin, 71, 139–140
  session hijacking of Tsutomu Shimomura's computer, 139–140, 143
modifying password settings, 311–312
monitoring
  ports, 424–426, 429
  session hijacking with packet sniffer, 145
morphing attacks, 324
mutating attacks, 324
MyDoom worm, 373
MySQL, 251
  databases structure, 251
  INFORMATION_SCHEMA views, 251
  structure of, 251

N

NASL (Nessus Attack Scripting Language), 440
NAT (Network Address Translation), configuring, 347
NDS (Novell Directory Services), 450
need for penetration testing, 8–9
need-based persuasion, 52
Nessus, 440
NetCat, 217–218
  shoveling remote shells, 395–397
  surreptitious installation of, 74
NetRecon, 445
Netstat, 426–427
NetStumbler, 354
network architecture, securing, 237
network attacks, tools for performing, 557
network hardening, preventing DoS attacks, 494–496
network libraries, 264–265
network-based IDSs, 110
networks, remote access, case study, 342, 345–347
newsgroups, 87
NICNAME, 91
NMAP, 22, 102
  compiling, 105
  switches, 103–105
nonblind spoofing, 128
non-executable stack, preventing buffer overflows, 475
non-U.S. laws pertaining to hacking, 31
NOPs (No Operation) commands, 464
NovelFFS (Novel Fake File Server), 451
Novell NetWare, 450
  hacking, 451
  NDS, 450
  rights, 450–451

NSLookup, 89–92
NT/2000 Rootkit, 395
NULL scans, 99–100
Nutcracker, 298–299

O

O'Keefe, Brett, 27
Oak, 193
obtaining access stage of attacks, 14
obtaining telecommuter phone numbers, 61
OECD (Organization for Economic Co-Operation and Development), 24
offline viewing of websites, 79
omni-directional antennas, 350
one factor authentication, 279
opcode, 466
open source software characteristics, 11
OpenBSD systems, inadvertant DoS attacks on, 22
open-source vulnerability scanners, 219
operating systems
  fingerprinting, 106
  securing, 232–234
  stack fingerprinting, 106
  UNIX, salts, 282
Oracle databases, 250
Orphan pages, 79
OSSTMM (Open-Source Security Testing Methodology Manual), 35–39, 503
out-of-sequence packets, 130

P

packet sniffers, 281
  detecting session hijacking, 145
  Ethereal, monitoring session hijacking attacks, 147, 151–153
packets, out-of-sequence, 130
Pandora, 451
Passive reconnaissance, 78
passive scanners, Kismet, 356
passive session hijacking, 127
password cracking
  attacks, detecting, 305–308
  brute force attacks
    Brutus, 211
    detecting, 212, 215
    HTTP Brute Forcer, 211
    protecting against, 215–216
  case study, 316–318
  employee education, implementing, 315–316
  protecting against, 309–314
  salts, 282
  utilities, 556
    Boson GetPass, 302–303
    Hypnopædia, 299
    John the Ripper, 285–287
    L0phtcrack, 289–290, 294, 298
    Nutcracker, 298–299
    pwdump, 277
    pwdump3, 287–288
    RainbowCrack, 303–304
    Snadboy Revelation, 300
password hashing, 280
  on Microsoft systems, 282–283
  on UNIX systems, 284
password policies, 542
passwords. See also *password cracking*
  brute force attacks, 209
    detecting on SQL servers, 268
  encrypted, 281
  erasing, 280
  hidden field exploit, 207, 209
  securing on routers, 338–340
  testing for vulnerabilities, 328
patience as social engineer trait, 56–57
penetration tests
  preparing for, 16–17
  vendors, selecting, 14–16
Perl (Practical Extraction and Report Language), 187–188
permissions, UNIX, 445
  assigning to root user, 445
  elevation techniques, 446–447
persuasion
  authority-based persuasion, 53
  conformity persuasion, 50
  information-based persuasion, 54–55
  logic persuasion, 51–52
  need-based persuasion, 52
  reciprocation-based persuasion, 53–54
  similarity-based persuasion, 54
phishing, 67, 203–205
phone numbers of telecommuters, obtaining, 61

PHP (Hypertext Preprocessor), 251
PHP (Personal Home Page), 192
***Phrack Magazine*, 131**
physical access policies, 543
physical protection, implementing, 313–314
PictureClock.class, 194
Ping of Death attacks, 483–484
PipeUpAdmin, 449
plain-text passwords, 280
plug-ins, BO2K, 380–382
port scanning, 96–97
- ACK scans, 100
- case study, 118–122
- dumb scans, 101
- FIN scans, 100
- inadvertant DoS attacks, 22
- NMAP, 102
- NULL scans, 99–100
- SYN scans, 99
- TCP connect() port scans, 98
- Xmas-Tree scans, 101

ports
- monitoring, 424–426, 429
- on Microsoft SQL Server, 265

preparing
- for penetration testing, 16–17
- security policies, 544

preventing
- attacks on wireless networks, 359, 362
  - *MITM attacks, 361*
- backdoor applications, 432–433
- buffer overflows, 473–475
- DoS attacks, 493
  - *through application hardening, 497*
  - *through network hardening, 494–496*
- server attacks, 452–455
- Trojan horse applications, 432–433

privilege escalation on IIS web servers, 201
privileged exec mode, 302
Project Scope, 42
protecting
- against brute force attacks, 215–216
- against database attacks, 270–271
- against password-cracking attacks, 309–312
- against session hijacking, 167
- against social engineering attacks, 71

protocol exceptions, 481
PS/SQL (Procedural Language/Structured Query Language), 250
Public Role (Microsoft SQL Server), 263
pwdump, 277
pwdump3, 287–288

Q-R

querying
- Microsoft SQL Servers, 253
- Oracle databases, 250

rainbow tables, 303
RainbowCrack, 303–304
RAT tools, Tini, 389
RDBMSs (Relational Database Management Systems), 249
- Microsoft SQL Server, 252
  - *authentication, 261–262*
  - *guest accounts, 264*
  - *network libraries, 264–265*
  - *ports, 265*
  - *Public Role, 263*
  - *querying, 253*
  - *sample databases, 264*
  - *Service Accounts, 263*
  - *structure of, 252*
- MySQL, 251
- Oracle, 250
- vulnerabilities, testing, 254
  - *brute force attacks, 260*
  - *connection strings, 259*
  - *SQL injection, 256*
  - *system stored procedures, 257*

reassessment principle, 24
reciprocation-based persuasion, 53–54
reconnaissance attacks, 13
- active host reconnaissance
  - *NSLookup/Whois lookups, 89–92*
  - *SamSpade, 92–94*
  - *Visual Route, 95*
- footprinting, 108

registers, 462
- ESP registers, 462
- general-purpose, 462
- special-purpose application, 464

regression testing, 7
regulatory laws, 28
remote access to networks, case study, 342, 345–347

remote administration Trojans, Beast, 412
client configuration, 417–419, 423
server settings, 412–416
remote port monitoring, 429
remote shell utilities
NetCat, 217–218
shoveling, 395–397
remote-access policies, 542
remote-access Trojans, Donald Dick, 390, 393
remote-administration trojans, detecting, 423
required topics in security policies, 538
Results Analysis, 42–43
Revelation, 300
rights (Novell NetWare), 450–451
RIP Route Generator, 330
risk analysis, 12
risk assessment, performing, 536
company acceptance, gaining, 538
cost of protecting assets, calculating, 537–538
threats to assets, identifying, 537
rogue APs, detecting on wireless networks, 358
root access, obtaining on Linux-based machines
irix-login.c exploit, 447
rpc.statd exploit, 446–447
stack-smashing exploit, 446
root user permissions, assigning, 445
rootkits, 395, 447
case study, 456–457
for Microsoft Windows, 450
routers, 324
CDP
disabling, 338
testing for vulnerabilities, 324–326
HTTP service
disabling, 338
testing for vulnerabilities, 326
passwords
securing, 338–340
testing for vulnerabilities, 328
routing protocols, testing for vulnerabilities, 329–332
securing, 338
routing protocols, testing for vulnerabilities, 329–332
rpc.statd exploit, 446–447
RSE (reverse social engineering), 70
Russinvoch, Mark, 429

S

SAINT (Security Administrators Integrated Network Tool), 441
salts, 282
SAM file, 276
sample databases on Microsoft SQL Server, 264
sample Executive Summary, 41
sample penetration test report, 524–529
SamSpade, 92–94
Sanfilippo, Salvatore, 101
Sapphire worm, 376–377
SARA (Security Auditor's Research Assistant), 442–443
Sarbanes-Oxley Act, 30
Sasser worm, 9, 377
scan detecting, 109
examples, 112
FIN scans, 115
NULL scans, 115
OS guessing scans, 117–118
SYN scans, 114
TCP Connect() scans, 113
Xmas tree scans, 115
scanners, 430–431
scanning stage of attacks, 14
scope of testing, defining, 6, 36
scripting languages
ASP, 188–190
JavaScript, 185–186
JScript, 186
Perl, 187–188
PHP, 192
VBScript, 186–187
script-kiddies, 11
securing
firewalls, 337
network architecture, 237
operating system, 232–234
routers, 338–340
web server applications
Apache, 236
IIS, 234–236
website design, 236
security policies, 7
authorship, 540
backup policies, 543
cost of protecting assets, calculating, 537–538

creating, 544
disaster recovery policies, 543
e-mail policies, 541
gaining company acceptance, 538
Internet policies, 541
password policies, 542
physical access policies, 543
remote access policies, 542
required topics, 538
threats to assets, identifying, 537
trust model, 540
types of, 541
selecting penetration testing vendor, 14–16
semi-directional antennas, 350
server attacks
detecting, 452
preventing, 452–455
server penetration, tools for performing, 562–565
server-based Java, 194–195
servers, honeypots, 17
service accounts, on Microsoft SQL Server, 263
session hijacking, 36, 127
ACK storms, 137–138
blind spoofing, 129
case study, 168–169, 172–173
detecting, 143–145
with packet sniffers, 145
detecting with Cisco IDS, 153–156, 162–164
Kevin Mitnick's attack on Tsutomu Shimomura's computer, 139–140, 143
monitoring with Ethereal, 147, 151–153
nonblind spoofing, 128
protecting against, 167
TCP sequence prediction, 130–131
utilities, 553
Hunt, 134–135
Juggernaut, 131–133
T-Sight, 136
watching, 164–165
TTY-Watcher, 136
session replays versus session hijacking, 128
Shimomura, Tsutomu, 139
shoveling remote shells, 395–397
showcode.asp, 200–201
signature-based IDSs, evading, 323
detecting DoS attacks, 490–492
signatures, detecting session hijacking, 155–156, 160, 164
similarity-based persuasion, 54
Simon, William, 71
simplex connection hijacking, 132
single-server e-commerce architecture, 198
Site Security Handbook, 535
Slammer worm, 376–377
SMBdie, 498
Smith, David, 27
Smurf amplifiers, 485
Smurf attacks, 484–485
Snadboy Revelation, 300
social engineering, 6, 36, 308
behavioral profiling, 55
case study, 72–75
defending against, 71
e-mail impersonation, 64–68
employees, impersonating,60
tech support, 61–62
end-users, impersonating, 69
human-based, 50
customers, impersonating, 69–70
persuasion, types of
authority-based persuasion, 53
conformity persuasion, 50
information-based persuasion, 54–55
logic persuasion, 51–52
need-based persuasion, 52
reciprocation-based persuasion, 53–54
similarity-based persuasion, 54
RSE, 70
technology-based, 49
third parties, impersonating, 62–64
traits necessary for
confidence, 57–58
patience, 56–57
possessing inside knowledge, 59
trust, 58
user group meetings, 88
web page spoofing, 203–205
wetware, 50
witness consultants as coaches, 58
sockets, 196
software, open source
characteristics of, 11
SOX (Sarbanes-Oxley) Act, 30
spacefiller viruses, 369
SPAN (Switched Port Analyzer), 129
special-purpose application registers, 464
Spendor Datapool, 486
spoofed e-mail messages, 64

SQL (Structured Query Language), 247
brute force server attacks, case study, 272, 275–277
commands, 249
master database, 252
servers
brute force attacks, detecting, 268
system stored procedures, 269
SQL injection, 236, 270
testing vulnerability to, 256
SQL Slammer worm, 265, 376–377
SQLPing2, 256
SSIDs (service set identifiers), 351
SSL infinite loops, 199
stack fingerprinting, 106
stack smashing exploit, 446
stacks, 461
non-executable, preventing buffer overflows, 475
stages of attacks
erasing evidence, 14
maintaining access, 14
obtaining access, 14
reconnaissance, 13
scanning, 14
standards for wireless networks, enforcing, 362
stored procedures, 257, 269
extended, 272
STP (Spanning Tree Protocol)
hardening switches against attacks, 341
testing switches for vulnerabilities, 334
structure
of Microsoft SQL Servers, 252
of MySQL databases, 251
of Oracle databases, 250
StumbVerter, 354
SubSeven, 398, 401–403, 406, 409–410
superusers, assigning permissions to, 445
susceptibility of databases to attack, 249
switches, 111–112
ARP attacks, hardening against, 341
MAC table flooding attacks, hardening against, 341
NMAP, 103–105
STP attacks, hardening against, 341
testing for vulnerabilities, 333
via ARP attacks, 335
via MAC table flooding, 335
via STP, 334
via VLAN hopping, 333–334
via VTP attacks, 336–337
VLAN hopping, 341
VTP attacks, hardening against, 342
symptoms of session hijacking, 143–145
SYN floods, 481, 486
SYN scans, 99
system log files, detecting password-cracking attacks, 306
system stored procedures, 257–258
system tablespace, 250
sysxlogins, 261–262

T

tablespaces, 250
TCP
embryonic connections, 140
sequence prediction, 130–131
session hijacking, 36
TCP Connect() scans, 98
TCPView, 429
tech support personnel, impersonating, 61–62
technology-based social engineering, 49
telecommuters, remote access policy, 542
Teleport Pro, 81
Telnet sessions, hijacking example, 135
temporal response analysis, 107
Ten Commandments of Computer Ethics, 22–23
testing
RDBMS for vulnerabilities, 254
brute force attacks, 260
connection strings, 259
SQL injection, 256
system stored procedures, 257
routers for vulnerabilities, 324
CDP, 324–326
HTTP service, 326
passwords, 328
routing protocols, 329–332
switches for vulnerabilities, 333
via ARP attacks, 335
via MAC table flooding, 335
via STP, 334
via VLAN hopping, 333–334
via VTP attacks, 336–337
testing reports
Appendixes, 44

Executive Summary section, 41, 44
Project Scope section, 42
Results Analysis section, 42–43
tests
black-box tests, 6
crystal-box, 6
gray-box, 6
white-box tests, 6
third-party impersonation, 62–64
threats, 7
availability threats, 7
confidentiality threats, 7
integrity threats, 7
tiered e-commerce architecture, 198
tiger teams, 5
time bomb viruses, 369
Tini, 389
Titanic syndrome, 8
TMRC (Tech Model Railroad Club), 5
TOE (target-of-evaluation), 6
tools
backdoor applications, 560–562
for attempting buffer overflows, 566
for attempting DoS attacks, 566–569
Datapool, 486
Hgod, 489
Jolt2, 488
for performing database attacks, 556
for performing network attacks, 557
for performing server penetration, 562–565
for performing web server attacks, 554–555
for wireless network penetration, 559
hacking tools, availability of, 10
host reconnaissance, 547, 550–553
password cracking, 556
session hijacking, 553
vulnerability scanners
ISS, 444
Nessus, 440
NetRecon, 445
SAINT, 441
SARA, 442–443
traits of social engineers
confidence, 57–58
patience, 56–57
possessing inside knowledge, 59
trust, 58
Trojan applications, 37, 67, 367, 560–562
Beast, 412
client configuration, 417–419, 423
gaining access with, case study, 433–436
server settings, 412–416
BO2K, 378–388
plug-ins, 380–382
Brown Orifice, 411–412
detecting, 423
Donald Dick, 390, 393
scanner software, 430–431
SubSeven, 398–403, 406, 409–410
trust as social engineer trait, 58
trust model, 540
T-Sight, 136
T-SQL (Transact-SQL), 253
TTY Watcher, 136
tunneling
ICMP tunneling, 323
Loki ICMP tunneling, 322
two-factor security, 279
Type-1 wireless, 353

U

unallocated MAC addresses, detecting on wireless networks, 359
unethical practices, free penetration testing, 32
UNIX
elevation techniques, 446
irix-login.c, 447
rpc.statd exploit, 446–447
stack smashing exploit, 446
password-cracking utilities, Nutcracker, 298–299
password hashing, 284
permissions, 445
assigning to root user, 445
rootkits, 447
salts, 282
UrlScan, 235–236
USA PATRIOT act, 30
USENET newsgroups, 87
user accounts, locking out, 311
user exec mode, 302
user group meetings, 88
utilities

- backdoor applications, 560, 562
- CleanIISLog, 222
- execiis-win32.exe, 221
- for attempting buffer overflows, 566
- for attempting DoS attacks, 566–569
- for attempting network attacks, 557
- for server penetration, 562–565
- for wireless network penetration, 559
- fport, 428
- host reconnaissance, 547, 550–553
- IIS Xploit, 221
- IntelliTamper, 222
- NetCat, 217–218
- Netstat, 426–427
- packet sniffers, 281
- password-cracking
  - *Boson GetPass, 302–303*
  - *Hypnopædia, 299*
  - *John the Ripper, 285–287*
  - *L0phtcrack, 289–290, 294, 298*
  - *Nutcracker, 298–299*
  - *Pwdump3, 287–288*
  - *RainbowCrack, 303–304*
  - *Snadboy Revelation, 300*
- session hijacking, 553
- TCPView, 429
- vulnerability scanners, 218
  - *commercial, 219*
  - *open-source, 219*
- Xprobe2, 107

V

VBScript, 186–187

vendors of penetration testing, selecting, 14–16

views (MySQL), INFORMATION_SCHEMA, 251

viruses, 9, 368. ***See also*** **worms**
- BugBear, 372–373
- Chernobyl, 369
- construction kits, 369
- Melissa, 371
- Sasser virus, 9
- scanner software, 430–431

Visual Route, 95

VLAN hopping
- hardening switches against, 341
- testing switches for vulnerabilities, 333–334

VTP attacks
- hardening switches against, 342
- testing switches for vulnerabilities, 336–337

vulnerabilities, 7
- of Apache HTTP Web Servers, 199
- of databases to attack, 249
- of IIS web servers, 199–200
  - *buffer overflows, 202–203*
  - *privilege escalation, 201*
  - *showcode.asp, 200–201*
- of RDBMSs, testing, 254–260

vulnerability scanners, 218, 439
- commercial, 219
- ISS, 444
- Nessus, 440
- NetRecon, 445
- open-source, 219
- SAINT, 441
- SARA, 442–443
- tests performed, 440
- Whisker attacks, detecting, 228–231

W

W32.CIH.Spacefiller, 369

W32.Slammer worm, 376–377

W32/BugBear virus, 372–373

W32/Klez worm, 373, 375

war dialing, 86

war driving, 353

war flying, 353

war pedaling, 353

war sailing, 353

war walking, 353

web attacks
- detecting, 225
- directory traversals, detecting, 226–228

web languages
- ASP, 188–190
- CGI, 191
- ColdFusion, 193
- DHTML, 181
- file extensions, 178
- HTML, 179

Java, 193
client-based, 194
server-based, 194–195
JavaScript, 185–186
JScript, 186
Perl, 187–188
PHP, 192
VBScript, 186–187
XHTML, 184
XML, 183–184
web pages
hidden fields, exploiting, 207–209
spoofing, 203–205
tools for performing, 554–555
web servers
Apache, 11
***applications, securing**, 236*
vulnerabilities, 199
attacks on, tools for performing, 197, 554–555
IIS
***applications, securing**, 234–236*
vulnerabilities, 199–203
web-based authentication attacks, 197
websites, 196–197
configuring NAT, 347
downloading for offline viewing, 79
Google.com as hacking tool, 224
OECD, 24
orphan pages, 79
securing, 236
sockets, 196
WEP (Wired Equivalency Protocol), 9, 352
WEPCrack, 357
wetware, 50
Wget, 79
Whisker attacks, detecting, 228–231
white-box tests, 6
white-hat hackers, 5
Whois lookups, 89–92
Windows
elevation techniques, 449
HK exploit, 449
PipeUpAdmin, 449
hashing, 282
password-cracking utilities, Hypnopædia, 299
privilege escalation, example of, 471
rootkits, 450
wireless networks
802.1x port security, 352
access points, 350
attacks
detecting, 357
preventing, 359–362
DoS attacks, detecting, 358
DSSS, 350
history of, 349–350
IPSec, 353
MAC address spoofing, detecting, 358
MAC filtering, 352
penetration testing tools, 559
AiroPeek NX, 357
AirSnort, 357
DStumbler, 355
GPSMAP, 356
Kismet, 355
NetStumbler, 354
StumbVerter, 354
WEPCrack, 357
rogue APs, detecting, 358
security, 351
case study, 363–365
SSIDs, 351
standards, enforcing, 362
Type-1, 353
unallocated MAC addresses, detecting, 359
war driving, 353
WEP, 352
witness consultants as social engineering coaches, 58
WLANs (wireless LANs), 9
worms, 9, 368
"I Love You," 370
Blaster, 375–376
MyDoom, 373
Sasser, 377
W32/Klex, 373–375

X-Y-Z

XHTML, 184
Xmas-Tree scans, 101
XML, 183–184
Xprobe2, 107

zero-day exploits, 8
zones, 90